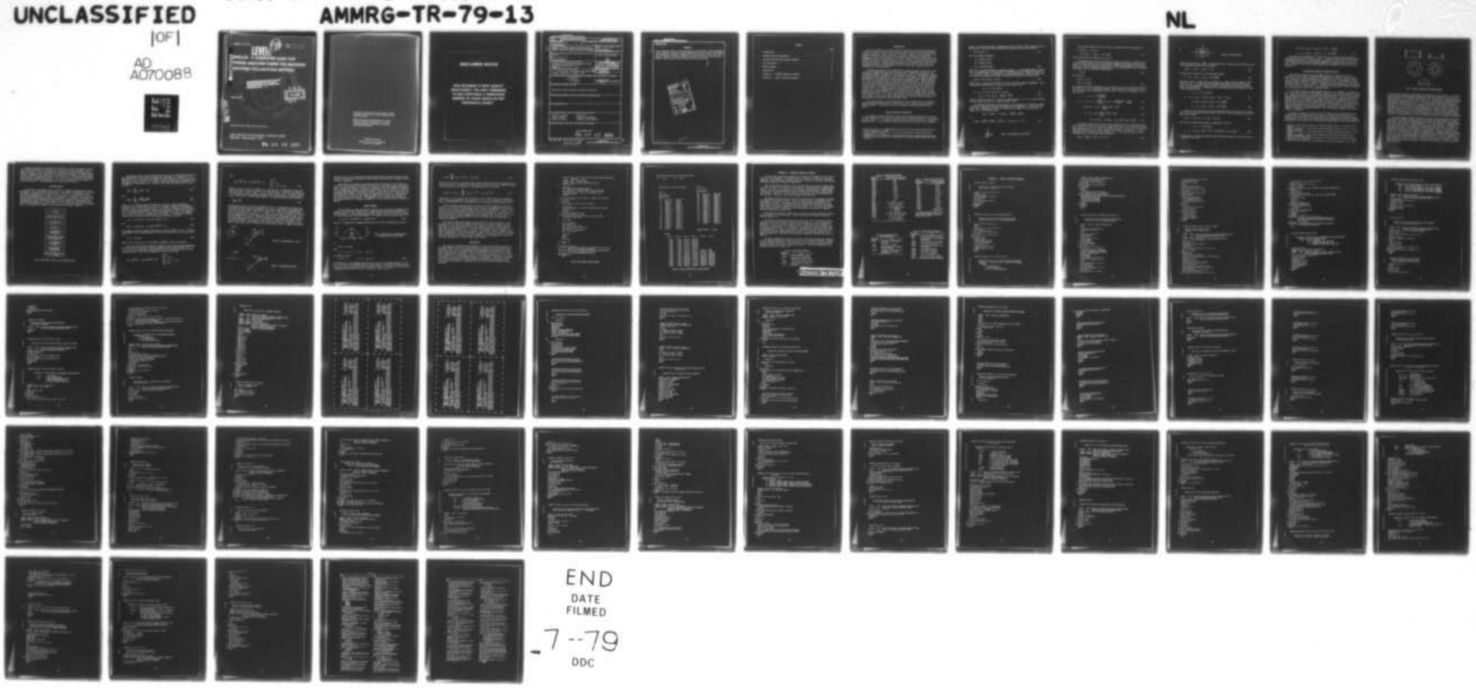
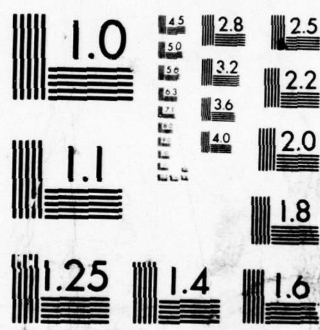
AD-A070 088 ARMY MATERIALS AND MECHANICS RESEARCH CENTER WATERTOWN MA F/G 9/2
MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS USING THE MODIFIED --ETC(U)
MAR 79 P G TRACY
UNCLASSIFIED AMMRG-TR-79-13 NL

1 OF 1
AD
A070088



END
DATE
FILMED
7--79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AMMRC TR 79-13

AD

LEVEL II

12

AD A070088

MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS USING THE MODIFIED MAPPING-COLLOCATION METHOD

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

March 1979

Approved for public release; distribution unlimited.

ARMY MATERIALS AND MECHANICS RESEARCH CENTER
Watertown, Massachusetts 02172

79 06 19 002

DISCLAIMER NOTICE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AMMRG-TR-79-13	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS USING THE MODIFIED MAPPING-COLLOCATION METHOD. (6)		5. TYPE OF REPORT & PERIOD COVERED (9) Final Report.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Peter G. Tracy		8. CONTRACT OR GRANT NUMBER(s) (16)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Army Materials and Mechanics Research Center Watertown, Massachusetts 02172 DRXMR- TM		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS D/A Project: 1T161101A91A AMCMS Code: 611101.91A0011 Agency Accession: DA OE4810
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Materiel Development and Readiness Command, Alexandria, Virginia 22333		12. REPORT DATE (11) March 1979
		13. NUMBER OF PAGES 53
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 59p.		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer programs
Stress analysis
Complex variables
Elasticity
Numerical techniques
Boundary value problems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(SEE REVERSE SIDE)

79 06 19 002

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

403 105

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

LB

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Block No. 20

ABSTRACT

A computer code for the solution of a wide variety of plane isotropic stress problems using the modified mapping-collocation method is presented. The mathematical approach, capabilities, and structure of the code are examined. Finally, a sample problem is solved to illustrate the ease with which stress solutions can be obtained utilizing the code.

1473B

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

CONTENTS

INTRODUCTION

The stress analysis for plane problems in elasticity has been handled using a variety of methods, which include finite element and finite difference methods, integral equations, integral transform techniques, complex variable methods, and boundary collocation. One particularly versatile technique is the modified mapping-collocation (MMC) method[1] and the MMC method with partitioning[2] developed at AMMRC by a team of investigators led by O. L. Bowie. This method combines the complex variable and conformal mapping techniques of Muskhelishvili[3] with boundary collocation methods. Finite element representations have been introduced into problems analyzed by the MMC method by Freese,[4] resulting in greater potential versatility for the MMC method.

Previous exploitation of the MMC method to plane problems of practical interest has been hampered by the time-consuming task of algebraic manipulation and subsequent computer programming of the equations used in a given problem. Since the equations used are similar in each problem, the computer code MMCLIB (Modified Mapping-Collocation LIBrary) has been written to eliminate the algebraic manipulation effort and greatly minimize the computer programming task. MMCLIB consists of a library of ninety Fortran subroutines and functions which, when combined in a Fortran main program written by the user, have the ability to quickly provide stress solutions for plane problems while retaining the versatility inherent in the MMC method. Furthermore, addition of new mapping functions and series representations can be accomplished with a minimum of effort, which allows the expansion of the code's capabilities. The versatility of the MMC method has been demonstrated in the solution of a wide variety of stress problems such as sheets with internal flaws, i.e., cracks and ellipses, sheets with edge cracks and elliptical notches, sheets with cracks emanating from elliptical internal flaws and edge notches, and circular rings with edge cracks.

The current report is not intended to be a user's manual but is intended to describe the capabilities of the code MMCLIB in the solution of plane isotropic stress problems. Information on the use of MMCLIB can be obtained from the author.

COMPLEX VARIABLE FORMULATION

The complex variable approach provides the mathematical formulation on which the MMC technique is based and results relating to the MMC method will be discussed in this section. For a complete treatment of the subject the reader is referred to Muskhelishvili.[3] A well-known method of describing the stresses in an isotropic

1. BOWIE, O. L., and NEAL, D. M. *A Modified Mapping Collocation Technique for Accurate Calculation of Stress Intensity Factors.* Int. J. Fracture Mech., v. 6, 1970, p. 199-206.
2. BOWIE, O. L., FREESE, C. E., and NEAL, D. M. *Solutions of Plane Problems of Elasticity Utilizing Partitioning Concepts.* J. Appl. Mech., v. 95, 1970, p. 767-772.
3. MUSKHELISHVILI, N. I. *Some Basic Problems of the Mathematical Theory of Elasticity.* Noordhoff, Groningen, Holland, 1953.
4. FREESE, C. E. *Collocation and Finite Elements - A Combined Method.* Army Materials and Mechanics Research Center, AMMRC TR 73-28, 1973.

elastic two-dimensional body is through the use of the Airy stress function, U(x,y). Specifically, equilibrium and compatibility conditions will be satisfied if

$$\nabla^2\nabla^2 U(x,y) = 0 \tag{1}$$

with the stresses defined by

$$\sigma_x = [\partial^2 U(x,y)]/\partial y^2$$

$$\sigma_y = [\partial^2 U(x,y)]/\partial x^2$$

$$\tau_{xy} = [-\partial^2 U(x,y)]/\partial x \partial y \tag{2}$$

where ∇^2 is the two-dimensional Laplacian operator. In the Muskhelishvili formulation, the coordinates of the problem are described using the complex variable $z = x + iy$ and the z-plane is referred to as the physical plane. The Airy stress function can then be represented using two analytic functions $\phi_z(z)$ and $\psi_z(z)$ by

$$U(x,y) = \mathrm{Re}\,[\bar{z}\,\phi_z(z) + X_z(z)] \tag{3}$$

where $\psi_z(z) = d\,X_z/dz$, Re means the real part, and bars complex conjugation. The stresses and displacements in terms of $\phi_z(z)$ and $\psi_z(z)$ are

$$\sigma_x + \sigma_y = 4\,\mathrm{Re}\,[\phi_z'(z)]$$

$$\sigma_y - \sigma_x + 2i\,\tau_{xy} = 2[\bar{z}\,\phi_z''(z) + \psi_z'(z)] \tag{4}$$

$$2\,G\,(u + iv) = k\phi_z(z) - z\overline{\phi_z'(z)} - \overline{\psi_z(z)} \tag{5}$$

with $G = E/2\,(1+\nu)$, and $k = (3-\nu)/(1+\nu)$ (plane stress), or $k = 3-4\nu$ (plane strain), E being Young's modulus and ν being Poisson's ratio.

An additional physical quantity which is used frequently in applying the MMC technique is the force acting on an arbitrary arc of material. If $X_n ds$ and $Y_n ds$ are x- and y-components of the force acting on an element of the arc ds (see Figure 1), it can be shown that

$$(X_n + i\,Y_n)ds = -i\,d\,\{\phi_z(z) + z\overline{\phi_z'(z)} + \overline{\psi_z(Z)}\}$$

or

$$\phi_z(z) + z\overline{\phi_z'(z)} + \overline{\psi_z(z)} = i\int (X_n + i\,Y_n)\,ds = f_1 + i\,f_2. \tag{6}$$

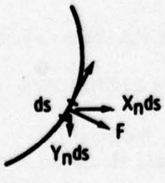


Figure 1. Forces acting on an arc of material.

The resultant moment about the origin of coordinates can be expressed as

$$M = \int (x\, Y_n - y\, X_n)\, ds$$

$$= \mathrm{Re}\, [X_z(z) - z\, \psi_z(z) - z\bar{z}\, \phi'(z)]_A^B \tag{7}$$

where A and B are the endpoints of the arc.

The formulation of the two-dimensional theory of elasticity in terms of analytic functions permits the use of conformal mapping techniques, providing the means to introduce curvilinear coordinates to describe the geometry in the physical plane. Use of conformal mappings leads to the introduction of the ζ or parameter plane with the conformal transformation

$$z = \omega(\zeta). \tag{8}$$

The functions

$$\phi(\zeta) = \phi_z\, (\omega(\zeta))$$

$$\psi(\zeta) = \psi_z\, (\omega(\zeta)) \tag{9}$$

are therefore analytic functions of ζ since $\phi_z(z)$ and $\psi_z(z)$ are analytic functions of z and $\omega(\zeta) = z$ is a conformal mapping. Substituting $\phi(\zeta)$ and $\psi(\zeta)$ for $\phi_z(z)$ and $\psi_z(z)$ and using relationships such as $\phi_z'(z) = \phi'(\zeta)/\omega'(\zeta)$, Equations 4, 5, 6, and 7 become

$$\sigma_x + \sigma_y = 4\, \mathrm{Re} \left[\frac{\phi'(\zeta)}{\omega'(\zeta)}\right]$$

$$\sigma_y - \sigma_x + 2i\, \tau_{xy} = 2\, \frac{\overline{\omega(\zeta)}}{(\omega'(\zeta))^2} \left(\phi''(\zeta) - \frac{\omega''(\zeta)\phi'(\zeta)}{\omega'(\zeta)}\right) + \frac{\psi'(\zeta)}{\omega'(\zeta)} \tag{10}$$

$$2G\, (u+iv) = k\, \phi(\zeta) - \frac{\omega(\zeta)}{\overline{\omega'(\zeta)}}\, \overline{\phi'(\zeta)} - \overline{\psi(\zeta)} \tag{11}$$

$$f_1 + if_2 = \phi(\zeta) + \frac{\omega(\zeta)}{\overline{\omega'(\zeta)}}\, \overline{\phi'(\zeta)} + \overline{\psi(\zeta)} \tag{12}$$

$$M = \mathrm{Re}\, [X(\zeta) - \omega(\zeta)\psi(\zeta) - \omega(\zeta)\, \overline{\omega(\zeta)}\, \phi'(\zeta)/\omega'(\zeta)]_A^B. \tag{13}$$

An extremely useful result of the work of Muskhelishvili is the application of the analytic continuation arguments summarized below. Let S^+ denote the region above the real axis ($y > 0$) and S^- the region below the real axis ($y < 0$) and further assume that the elastic body occupies the region S^+ (see Figure 2). The functions $\phi_z(z)$ and $\psi_z(z)$ are thus defined in S^+. The functions $\phi_z(z)$ shall be continued into S^- by the following definition:

$$\phi_z(z) = -z\bar{\phi}_z'(z) - \bar{\psi}_z(z) \quad \text{for} \quad z \in S^-, \tag{14}$$

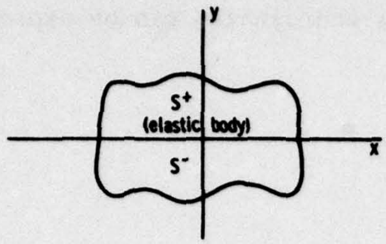


Figure 2. Analytic continuation.

with the notation $\bar{f}(z) = \overline{f(\bar{z})}$. Noting that if $z \in S^-$, then $\bar{z} \in S^+$ and taking the complex conjugate of Equation 14 one finds

$$\psi_z(z) = -\bar{\phi}_z(z) - z\,\phi_z'(z), \quad z \in S^+. \tag{15}$$

Substitution of Equation 15 into Equation 6 gives

$$f_1 + if_2 = \phi_z(z) - \phi_z(\bar{z}) + (z-\bar{z})\,\overline{\phi_z'(z)}. \tag{16}$$

Evaluation of this equation for z with Im(z) = 0, where Im means imaginary part, gives $f_1 + if_2 = 0$. Thus, the real axis has been defined as load-free, while $\psi_z(z)$ has been eliminated from the analysis leaving only the analytic function $\phi_z(z)$ to be determined. If Equation 15 is also substituted into Equations 4 and 5 the following results:

$$\sigma_x + \sigma_y = 4\,\mathrm{Re}\,[\phi_z'(z)]$$

$$\sigma_y - \sigma_x + 2i\,\tau_{xy} = 2\,[(\bar{z}-z)\,\phi_z''(z) - \bar{\phi}_z'(z) - \phi_z'(z)] \tag{17}$$

$$2\,G(u+iv) = k\phi_z(z) + \phi_z(\bar{z}) + (\bar{z}-z)\,\overline{\phi_z'(z)}$$

$$f_1 + if_2 = \phi_z(z) - \phi_z(\bar{z}) + (z-\bar{z})\,\overline{\phi_z'(z)} \tag{18}$$

Similarly the function $\phi_z(z)$ can be continued from the exterior of the unit circle (S^+) to the interior of the unit circle (S^-) using the following definition due to Kartzivadze[5]

$$\phi_z(z) = -z\,\bar{\phi}_z'\,(1/z) - \bar{\psi}_z\,(1/z) \quad \text{for } z \in S^-. \tag{19}$$

This leads to expressions similar to Equations 17 and 18

$$\sigma_x + \sigma_y = 4\,\mathrm{Re}\,[\phi_z'(z)]$$

$$\sigma_y - \sigma_x + 2i\,\tau_{xy} = 2[(\bar{z} - 1/z) + 1/z^2\,\bar{\phi}_z'(1/z) + 1/z^2\,\phi_z'(z)] \tag{20}$$

5. KARTZIVADZE, I. N. *The Fundamental Problems of the Theory of Elasticity for the Elastic Circle.* Comp. Rend. de L'acad. Sc. de l'URSS, v. 20, 1943, p. 95-104.

$$2\,G(u+iv) = k\phi_z(z) + \phi_z(1/z) + (1/\overline{z} - z)\,\overline{\phi_z'(z)}$$

$$f_1 + if_2 = \phi_z(z) - \phi_z(1/\overline{z}) + (z - 1/\overline{z})\,\overline{\phi_z'(z)}. \qquad (21)$$

The evaluation of Equation 21 on the unit circle shows that traction-free conditions must exist on the unit circle. It should be noted that the definitions in Equations 19 and 13 for the analytic continuation of $\phi_z(z)$ from S^+ to S^- will need slight modification when conformal mapping is included in the analysis and $\phi(\zeta) = \phi_z(\omega(\zeta))$ is used due to the presence of the mapping function in the force Equation 12.

THE MODIFIED MAPPING-COLLOCATION METHOD

The modified mapping-collocation (MMC) method is a technique for two-dimensional elastic stress analysis which combines the complex variable formulation and conformal mapping arguments of Muskhelishvili[3] and the boundary collocation technique, retaining the attractive features of each. This method was first proposed by Bowie and Neal,[1] and was refined with the introduction of a partitioning plan by Bowie, Freese, and Neal.[2] The MMC method has subsequently been applied to a wide variety of problems,[6-11] illustrating its versatility. Refer to Figure 3 for illustrations of some of the problems solved.

In the MMC technique a simple mapping function as in Equation 7 is chosen to describe critical portions of the physical geometry, e.g., cracks, notches, cutouts, etc. Remaining portions of the geometry can be described by inversion of the plane mapping function from the physical to the parameter plane. Thus, the forms of the mapping functions are algebraically simple since the entire geometry need not be described by a single function, and greater versatility is obtained since a unique mapping function need not be obtained for each problem geometry. The specification in the parameter plane of those portions of the boundary not described by the mapping function may be algebraically complex, making direct specification of boundary conditions extremely difficult. For this reason a boundary collocation scheme is introduced.

In order to obtain $\phi(\zeta)$ and $\psi(\zeta)$ for the problem under consideration a representation for each function must be chosen. Generally, when analyticity is assured, power and Laurent series are chosen for representation, and the unknown coefficients of the series are determined using the boundary collocation arguments. The forms for $\phi(\zeta)$ and $\psi(\zeta)$ are substituted into Equations 10 to 13, and equations

6. BOWIE, O. L. *Methods of Analysis and Solutions of Crack Problems* in Mechanics of Fracture, v. 1, G. C. Sih, ed., Noordhoff, Leden, 1973.
7. BOWIE, O. L., and FREESE, C. E. *Analysis of Notches Using Conformal Mapping* in Mechanics of Fracture, v. 5, G. C. Sih, ed., Noordhoff, Leden, 1978.
8. BOWIE, O. L., and FREESE, C. E. *On the 'Overlapping' Problem in Crack Analysis.* Engineering Fracture Mechanics, v. 8, 1976, p. 373-379.
9. BOWIE, O. L., and FREESE, C. E. *Elastic Analysis for a Radial Crack in a Circular Ring.* Army Materials and Mechanics Research Center, AMMRC MS 70-3, 1970.
10. TRACY, P. G. *Elastic Analysis of Radial Cracks Emanating from the Outer and Inner Surfaces of a Circular Ring.* Engineering Fracture Mechanics, v. 11, 1979, p. 291.
11. TRACY, P. G. *Analysis of a Radial Crack in a Circular Ring Segment.* Engineering Fracture Mechanics, v. 7, 1975, p. 253-260.

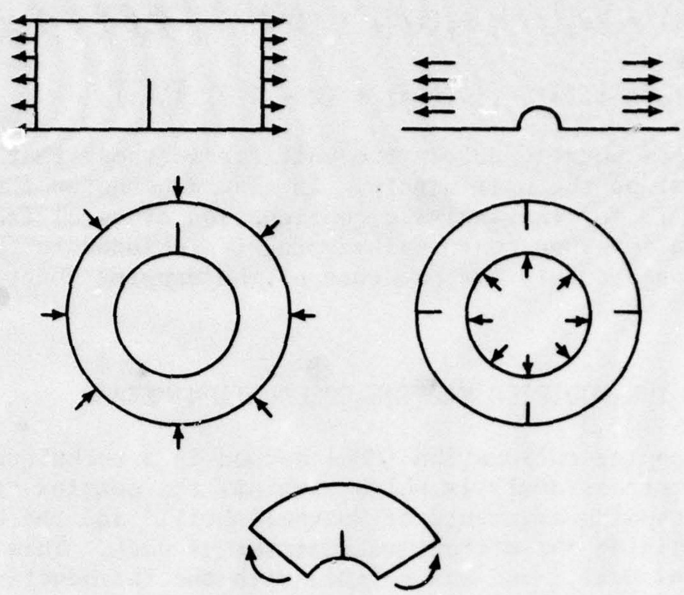


Figure 3. Examples of problems solved using MMC method.

can be written in terms of the unknown coefficients of $\phi(\zeta)$ and $\psi(\zeta)$. In situations where the representations for $\phi(\zeta)$ and $\psi(\zeta)$ are complete, collocation of f_1 and f_2 (Equation 12) is usually sufficient. In a region with a crack tip near a boundary, error in matching f_1 and f_2 may result, indicating the need for collocating stresses, Equation 10, as well as f_1 and f_2, particularly on the portion of the boundary near the crack tip. In cases where the representation may not be complete, the collocation of M, Equation 13, as well as forces and stresses may be required. A set of points on the boundaries is chosen at which to write equations such that the number of equations generated is greater than the number of unknown coefficients by at least a factor of two. A least-squares minimization of error is used to generate a linear system of equations in the unknown coefficients; the coefficients can then be determined by solving the system of equations.

The partitioning concepts introduced by Bowie, Freese, and Neal[2] extend the flexibility of the MMC technique to allow solutions for problems which are awkward with a single series representation for the stress functions. Such problems occur when, for example, singularities occur near the circles of convergence for the stress functions due to areas of high stress concentrations. To implement the partitioning plan, the geometry is subdivided into two or more regions. The functions $\phi_z(z)$ and $\psi_z(z)$ are defined in a subregion i by $\phi_{z,i}(z)$ and $\psi_{z,i}(z)$, along with a mapping for each region, if appropriate. Boundary conditions (Equations 10 to 13) are enforced along the boundary using the appropriate $\phi_{z,i}(z)$ and $\psi_{z,i}(z)$ for the collocation point at which equations are written. Additionally, at points on the common boundaries of zones $u+iv$ and $f_1 + if_2$ are equated to assure continuity of $\phi_z(z)$ and $\psi_z(z)$. Then the least-squares procedure is invoked and the resulting equations solved to obtain the function $\phi_{z,i}(z)$ and $\psi_{z,i}(z)$ in each subregion.

Once the solution is obtained it is necessary to assess the degree to which the boundary conditions are satisfied by $\phi_z(z)$ and $\psi_z(z)$ as obtained. This is done in two ways. First, the solution is recomputed, varying the number and location of points at which equations are written and the number of unknown coefficients to confirm that a wide range of these choices will yield the same results, thus assuring a convergent solution. As a second check, the boundary conditions are evaluated using the computed $\phi_z(z)$ and $\psi_z(z)$ and compared to the values required for solution. Good agreement will assure correct results.

THE CODE MMCLIB

The computer code MMCLIB has been written to enhance implementation of the MMC method. To retain maximum flexibility in the variety of problems to be solved, the code is constructed in a modular fashion. The user is required to write a Fortran main program which usually is 30 to 100 lines in length and uses the modules (Fortran subroutines and functions) of MMCLIB. In addition, the modular construction allows the implementation of additional mapping functions and series representation types as one need program only the function or representation and its first two derivatives into the code. Let us now consider the solution process, which is illustrated in Figure 4, and examine the features of MMCLIB.

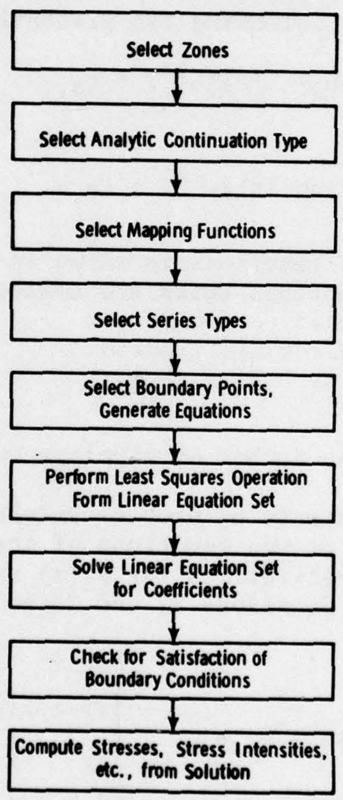


Figure 4. Block diagram of solution process using MMC method.

The problem geometry under consideration must first be subdivided into one or more subregions or zones with reflections as in Equation 14 or Equation 19, if appropriate. This will define the number and type (either $\phi_{z,i}(z)$ or $\psi_{z,i}(z)$) of functions needed to complete the solution. Each such function is defined by zone in MMCLIB by a subroutine call. At present two functional representations are available:

$$f_1(W) = \sum_{n=-N}^{M} a_n(W^n - W_o)^n \tag{22}$$

and

$$f_2(W) = \sum_{n=-N}^{M} \frac{a_n(W^n - W_o)^n}{W+i} \tag{23}$$

where a_n is an unknown complex coefficient, W is the (complex) parametric-coordinate or z-coordinate in the case of no mapping, M and N are nonnegative integers, and W_o is the expansion point. A function is defined by specifying W_o, M, N, the series representation, the material parameters of that zone, and any real or imaginary parts of a_n for even or odd powers of n which are assumed to be zero due to the stress symmetries of the problem. For each zone a mapping function may be chosen from the following two presently available

$$\omega_1(\zeta) = [(a_1+b_1)\zeta/2 + (a_1-b_1)\zeta^{-1}/2]e^{i\theta_1} + \zeta_{0,1} \tag{24}$$

and

$$\omega_2(\zeta) = i[b_2\cosh(i\zeta) - a_2\sinh(i\zeta)]e^{i\theta_2} + \zeta_{0,2}. \tag{25}$$

The action of these two mapping functions is shown in Figures 5 and 6. In addition, MMCLIB allows mapping functions which are compositions of $\omega_1(\zeta)$ and $\omega_2(\zeta)$, i.e.,

$$\omega(\zeta) = \omega_i[\omega_j(\zeta)] \tag{26}$$

where i and j vary from 1 to the number of mappings currently available.

One of the most tedious aspects of problem solving using the MMC method has been the need to algebraically reduce equations of the type Equations 10 to 13, 17, 18, 20, and 21 with the substitutions for $\phi_z(z)$ and $\psi_z(z)$ or $\phi(\zeta)$, $\psi(\zeta)$ of representations of the form of Equations 22 and 23 to the form

$$[a_{nr}\, f_{rr}^{(n)}(W) + a_{ni}\, f_{ri}^{(n)}(W)] = \mathrm{Re} \begin{cases} u+iv \\ f_1 + if_2 \\ \sigma_y - \sigma_x + 2i\,\tau_{xy} \\ 4\,\phi_z'(z) \end{cases}$$

and

$$[a_{nr}\ f_{ir}^{(n)}(W) + a_{ni}\ f_{ii}^{(n)}(W)] = \mathrm{Im} \begin{cases} u+iv \\ f_1 + if_2 \\ \sigma_y - \sigma_x + 2i\ \tau_{xy} \end{cases} \tag{27}$$

where $a_n = a_{nr} + i\ a_{ni}$, W is either ζ or z and the f's are functions of the position of the point at which equations are being written. Collocation of forms for functions such as $\alpha\phi_z(z)$, $\alpha\phi_z'(z)$, $\alpha\phi_z''(z)$, $\alpha\phi_z(\overline{z})$, $\alpha\overline{\phi}_z(z)$, etc., are each contained in a subroutine, and equations are generated by successive calls to these subroutines. Note that α is some complex number. The form assumed for each function is

$$\sum_n a_n\ f_n(W) \tag{28}$$

where presently $f_n(W)$ is either of Equation 22 or 23. Other forms for $f_n(W)$ can be added by simply inserting programming for $f_n(W)$, $[df_n(W)/dW]$, and $[d^2f_n(W)/d^2W]$ into the three appropriate subroutines. Many of the forms for the mapping functions such as $\omega(\zeta)$, $\omega(\overline{\zeta})$, $\omega'(\zeta)$, $\omega''(\zeta)$, $\overline{\omega}(\zeta)$, etc., are also contained in modules for use in generation of equations. It should be noted that if $\omega_i(\zeta)$, $\omega_i'(\zeta)$, $\omega_i''(\zeta)$ are programmed into the three appropriate subroutines, the mapping function $\omega_i(\zeta)$ can be included for analysis. Certain commonly used equations for problem solving have been programmed into modules using previously described subroutines. Thus, equations for forces, displacements, and stresses for real axis reflections, unit circle reflections, and no reflection are contained in modules. MMCLIB, therefore,

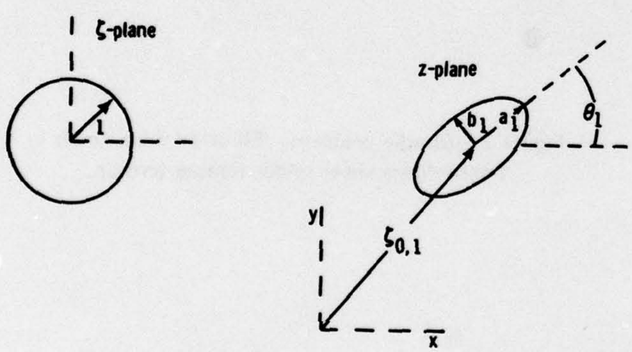


Figure 5. The mapping function $z = \omega_1(\zeta)$.

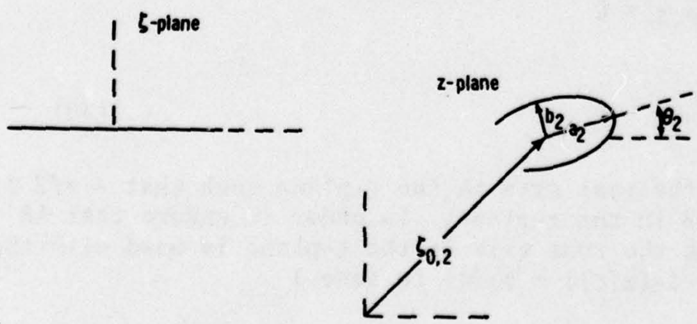


Figure 6. The mapping function $\omega_2(\zeta)$.

eliminates most of the repetitious equation derivation, allowing the user to concentrate on modelling the problem at hand without spending time in algebraic manipulation of equations.

After points are chosen for collocation and the equations in the unknown coefficients are generated and solved, the results of the computations can be examined. Any physical quantity can then be calculated using the computed coefficients and the appropriate modules to generate equations for the quantities of interest. The more common physical quantities of stress, displacements, and forces have been programmed into modules to facilitate the acquisition of results. Subroutines are also available to print out the computed coefficients and the input information about the various series in use. See Appendix A for the names and functions of the important modules of MMCLIB and Appendix B for the Fortran source listings for MMCLIB.

SAMPLE PROBLEM

To illustrate the ease with which solutions can be obtained using MMCLIB, the problem of an elliptical edge notch in a semi-infinite sheet under tension was solved (Figure 7). This problem was studied by Bowie and Freese[7] and their method of solution was duplicated using MMCLIB. The mapping function used for zone I is

$$z = \omega(\zeta) = i\ [(a+b)e^{-i\zeta/2} - (a-b)e^{-i\zeta/2}] \tag{29}$$

which is a composition of mapping functions of the type in Equations 24 and 25.

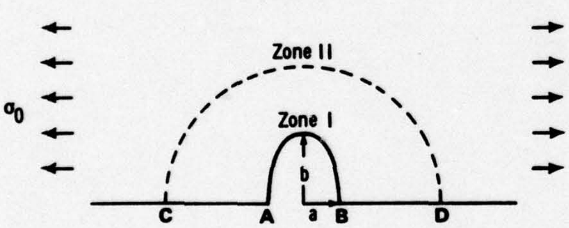


Figure 7. Sample problem: Elliptical edge notch in semi-infinite sheet under remote tension.

If

$$\omega(\zeta) = \omega_1(\omega_2(\zeta))$$

then

$$a_1 = a \quad b_1 = b \quad \theta_1 = 0 \quad \zeta_{0,1} = 0$$

in Equation 24 and

$$a_2 = b_2 = 1 \quad \theta = 0 \quad \zeta_{0,2} = 0 \tag{30}$$

in Equation 25. This mapping takes the real axis in the ζ-plane such that $-\pi/2 \le \mathrm{Re}\ \zeta \le \pi/2$ into the elliptical arc AB in the z-plane. In order to ensure that AB is traction-free, a reflection across the real axis in the ζ-plane is used with the following representation for $\phi_z(z) = \phi_z(\omega(\zeta)) = \phi_I(\zeta)$ in zone I

$$\phi_I(\zeta) = \sum_{n=1}^{N_1} [a_{2n-1}\, \zeta^{2n-1} + i\, a_{2n}\, \zeta^{2n}] \tag{31}$$

where the a's are real, ensuring stress symmetry about the physical imaginary axis. In zone II there is no mapping function and a reflection is used to ensure traction-free conditions on the real azis. The representation for $\phi_Z(z)$ is

$$\phi_{II}(z) = (1/4)\,\sigma_0 + \sum_{n=0}^{N_2} [a_{-2n-1}\, z^{-2n-1} + i\, a_{-2n}\, z^{-2n}] \tag{32}$$

ensuring $\sigma_x = \sigma_0$ for large z and the proper stress symmetry about the imaginary axis. Along the segment BD the condition $f_1 + if_2 = 0$ is collocated, and the condition of continuity of $f_1 + if_2$ and u+iv are collocated on the circular arc CD.

The Fortran main program used in the solution of this problem is shown in Figure 8 to illustrate the magnitude of the computer programming effort necessary to carry out a solution using the MMC method with the code MMCLIB. Lines 8 to 12 define the series and mapping functions to be used and lines 16 to 20 and 22 to 30 find points and collocate equations on the curves BD and DE, respectively. The printing of results of the computations are programmed into lines 34 to 40. It can be seen that programming effort to solve a problem using the MMC method is minimal with MMCLIB.

The computer generated output for this problem is shown in Figure 9. The maximum stress as computed by MMCLIB compares well with that computed by Bowie and Freese. The error calculation in this example is typical of many problems. Note that the right-hand sides of the equations generated are of the order of magnitude of the coordinates (i.e., unity). The amount of error compared to the order of magnitude of the values desired as shown in the first two columns of the error computation are usually considered adequate for accurate results. The third and fourth columns and fifth and sixth columns are the coordinates in zones one and two, respectively, at which errors were checked.

CONCLUSIONS

The computer code MMCLIB should permit a savings in the effort involved in obtaining stress solutions to plane problems of elasticity using the MMC method since the effort will be concentrated on the mathematical and physical concepts involved in solving the problems and not on mechanical activities such as algebraic manipulation of equations. Thus, problems whose solutions are not conceptually difficult to the experienced user of the MMC method but which were tedious to implement can be solved almost routinely using MMCLIB. Furthermore, research into further development of the MMC method can be performed much more easily since most of the analyst's attention can be focused on the new features being developed.

```
      DIMENSION X(50),Y(50),XP(50),YP(50),F1(50),F2(50),U(50),V(50),
     *          GPAR(5)
      COMMON   /MAT/  G(5),ETAM(5)
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      COMMON  /SOLN/  A(15401),IDIAG(176),PHI(176)
      DATA GPAR /5*0./
C
      CALL INIT
      READ 1000,IPOS,INEG,NPT,NST,AA,BB
      CALL SERINF(1,1,1,0,IPOS,0,1001,1,0.,0.,1.E+7,.3)
      CALL SERINF(2,1,1,INEG,0,1,1001,1,0.,0.,1.E+7,.3)
      CALL MAPINF(1,2,1.,1.,0.,0.,0.,1,AA,BB,0.,0.,0.)
      RZ=AMAX1(AA,BB)*2.
      IERR=0
C
  100 CALL POINTF(1,AA,0.,RZ,0.,NPT,1,1,1,GPAR,1,2,X,Y,XP,YP)
      DO 110 I=1,NPT
      F1(I)=0.
  110 F2(I)=0.
      CALL FDCOL(XP,YP,NPT,1,1,2,F1,F2,IERR)
C
      GPAR(1)=RZ
      CALL POINTF(1,RZ,0.,0.,RZ,NST,0,0,2,GPAR,0,2,X,Y,XP,YP)
      DO 120 I=1,NST
      F1(I)=0.
      F2(I)=Y(I)
      U(I)=X(I)*(ETAM(1)+1.)*.25
  120 V(I)=Y(I)*(ETAM(1)-3.)*.25
      CALL STITCH(1,2,2,2,3,2,0,NST,X,Y,F1,F2,U,V,IERR)
      IF(IERR.EQ.1)GO TO 130
C
      CALL FNDMAT
C
      PRINT 1002,AA,BB
      CALL PRTCOF(12)
      CALL ZERO(1,2,1,2,3,0)
      CALL PHIP(.0001,0.,4./AA,0.,1)
      CALL PLACE(1,1,1,1,1,1.)
      CALL EVAL(1,1,1,STRMAX,SDUM)
      PRINT 1003,STRMAX
C
      PRINT 1001
      IERR=1
      NST=6
      NPT=10
      GO TO 100
C
  130 PRINT 1004
      STOP
C
 1000 FORMAT(4I5,2E10.4)
 1001 FORMAT(7HOERRORS//4X,7HERROR 1,3X,7HERROR 2,3X,6HCOR 1R,4X,
     *       6HCOR 1I,4X,6HCOR 2R,4X,6HCOR 2I)
 1002 FORMAT(45H1ELLIPTICAL EDGE NOTCH IN SEMI-INFINITE SHEET//
     *       4H A =,F8.4,10X,3HB =,F8.4/////)
 1003 FORMAT(////17H MAXIMUM STRESS =,F8.4////)
 1004 FORMAT(1H1)
      END
```

Figure 8. Main program for sample problem.

ELLIPTICAL EDGE NOTCH IN SEMI-INFINITE SHEET

A = .1250 B = .5000

COEFFICIENTS IN PHI,PSI FUNCTIONS

ZONE NO. 1

PHI FUNCTION
SERIES TYPE 1

POW	REAL COEF	IMAG COEF
0	.00000	.00000
1	.30070+00	.00000
2	.00000	.39187-01
3	-.40076-01	.00000
4	.00000	-.85981-02
5	.38902-02	.00000
6	.00000	-.16099-03
7	.22604-03	.00000
8	.00000	-.17401-04
9	.51200-04	.00000
10	.00000	.11194-04
11	.41939-05	.00000
12	.00000	.13837-05
13	.13354-05	.00000
14	.00000	.13379-05
15	-.37054-06	.00000
16	.00000	-.36768-07
17	.35488-07	.00000
18	.00000	.55372-07
19	-.23934-07	.00000
20	.00000	-.44828-08

ZONE NO. 2

PHI FUNCTION
SERIES TYPE 1

POW	REAL COEF	IMAG COEF
-20	.00000	.34927-05
-19	-.19418-04	.00000
-18	.00000	-.47040-04
-17	.63436-04	.00000
-16	.00000	.32696-04
-15	.17873-04	.00000
-14	.00000	.47948-04
-13	-.11984-04	.00000
-12	.00000	.77529-04
-11	-.14141-03	.00000
-10	.00000	-.14408-03
-9	.19836-03	.00000
-8	.00000	.28798-03
-7	-.67002-03	.00000
-6	.00000	-.12855-02
-5	.34014-02	.00000
-4	.00000	.63706-02
-3	-.19156-01	.00000
-2	.00000	-.33425-01
-1	.16262+00	.00000
0	.00000	.22439+00

MAXIMUM STRESS = 9.6224

ERRORS

ERROR 1	ERROR 2	COR 1R	COR 1I	COR 2R	COR 2I
-.0003	-.0006	.1571+01	.1888+00		
-.0003	-.0000	.1571+01	.3640+00		
.0005	-.0000	.1571+01	.5239+00		
.0001	.0000	.1571+01	.6688+00		
-.0003	.0003	.1571+01	.7998+00		
-.0001	.0001	.1571+01	.9187+00		
.0001	-.0003	.1571+01	.1027+01		
.0000	-.0001	.1571+01	.1126+01		
.0001	.0004	.1571+01	.1217+01	.1000+01	.0000
.0000	-.0000	.1571+01	.1217+01	.1000+01	.0000
.0000	.0001	.1287+01	.1209+01	.9511+00	.3090+00
-.0000	-.0001	.1287+01	.1209+01	.9511+00	.3090+00
-.0001	-.0001	.9949+00	.1185+01	.8090+00	.5878+00
.0000	.0000	.9949+00	.1185+01	.8090+00	.5878+00
.0001	.0001	.6866+00	.1150+01	.5878+00	.8090+00
-.0000	-.0001	.6866+00	.1150+01	.5878+00	.8090+00
-.0000	-.0001	.3542+00	.1114+01	.3090+00	.9511+00
.0000	.0001	.3542+00	.1114+01	.3090+00	.9511+00
-.0000	.0001	.2980-07	.1099+01	.1589-07	.1000+01
.0000	-.0000	.2980-07	.1099+01	.1589-07	.1000+01

Figure 9. Computer generated output for sample problem.

APPENDIX A. IMPORTANT MODULES OF MMCLIB

In this section the more commonly used subroutines of MMCLIB will be listed and their uses explained. The subroutines are broken up by function into five groupings: Zone Definition, Mapping Functions, Function Collocation, Specialized Equation Generation, and Results Computation.

The Zone Definition routines allow the definition of functions (either $\phi_Z(z)$ or $\psi_Z(z)$) in each zone to be used in the analysis and definition of mapping functions in each zone. In addition, the functional representations, material properties, and any unknowns needed but not included in the functional forms chosen are defined. Note that all zone definition subroutine calls are preceded by a call to INIT. Table A-1 lists these routines and their uses.

The Mapping Function routines allow the calculation of coordinates in either the parametric or physical plane. These routines are used in determination of coordinates of points in one plane when the corresponding coordinates in the other plane are known. The Mapping Function routines and the forms of the mapping function which they compute are listed in Table A-2. Note that some subroutines return a number of functional forms depending on the number of mapping functions defined in a given zone.

The Function Collocation subroutines are used to write equations for the unknown parameters of the functions. Several Function Collocation routines are listed in Table A-3.

The Specialized Equation Generation routines are used to generate multiple equations for the most commonly used physical quantities. These routines are frequently used since in most solutions using the MMC method, the equations collocated on a given part of the boundary of the configuration being studied are of the same type and thus it is convenient to generate all these equations with one subroutine call. Specialized Equation Generation routines make use of Function Collocation and Mapping Function routines to generate the equations for physical quantities. The Specialized Equation Generation routines are listed in Table A-4.

The Result Computation routines are used to print the coefficients computed in the solution and physical quantities which can be obtained using the solution for the unknown coefficients. The routines are also used to print the various parameters input in the zone information routines. The Result Computation subroutines are listed in Table A-5.

Table A-1. ZONE DEFINITION ROUTINES

Subroutine	Use
INIT	Initialize MMCLIB variables
SERINF	Define functions and material properties to be used in zone
MAPINF	Define mapping function in zone
CONINF	Define unknown not contained in functions

Table A-2. MAPPING FUNCTION ROUTINES

Subroutine	Mapping Function Calculated
MAPB	$\omega(\zeta)$
MAPBAR	$\overline{\omega(\overline{\zeta})}$
MAPBR	$\overline{\omega}(\zeta)$
MAPBR1	$\overline{\omega}(1/\zeta)$
MAPB1	$\omega(1/\overline{\zeta})$
MAPPBR	$\overline{\omega'(\zeta)}$
MAPPB1	$\overline{\omega}'(\zeta)$
MAPPB2	$\overline{\omega}'(1/\zeta)$
MAPPP1	$\omega''(\zeta)$
MAPP1	$\omega'(\zeta)$
MAP1	$\omega(\zeta)$
MAPB12	$1/\overline{z}$, $\omega(1/\overline{\zeta})$, $\omega_2(\omega_1(1/\overline{\zeta}))$
MAPB21	$\overline{z}$, $\omega(\overline{\zeta})$, $\omega_2(\omega_1(\overline{\zeta}))$
MAPPR	1, $\omega'(\overline{\zeta})$, $\omega_2'(\omega_1(\zeta))\ \omega_1'(\zeta)$
MAPPRB	1, 1, $\overline{\omega_2'(\omega_1(\zeta))}$
MAPPRP	1, 1, $\omega_2'(\omega_1(\zeta))$
MAPP12	1, $\omega'(1/\overline{\zeta})$, $\omega_2'(\omega_1(1/\overline{\zeta}))\ \omega_1'(1/\overline{\zeta})$
MAPP21	1, $\omega'(\overline{\zeta})$, $\omega_2'(\omega_1(\overline{\zeta}))\ \omega_1'(\overline{\zeta})$
MAPPER	z, $\omega(\zeta)$, $\omega_2(\omega_1(\zeta))$
	z, $\omega^{-1}(\zeta)$, $\omega_1^{-1}(\omega_2^{-1}(\zeta))$

Table A-3. FUNCTION COLLOCATION ROUTINES

Subroutine	Function Collocated
PHI	$\alpha f\,(z)$
PHIB	$\alpha f\,(\overline{z})$
PHIBP	$\alpha\overline{f}'(z)$
PHIBR	$\alpha\overline{f\,(z)}$
PHIB1	$\alpha f\,(1/\overline{z})$
PHIB1P	$\alpha f'(1/z)$
PHIP	$\alpha f'(z)$
PHIPB	$\alpha\overline{f'(z)}$
PHIPP	$\alpha f''(z)$
PHIPPB	$\alpha\overline{f''(z)}$
PHIBP1	$\alpha f'(\overline{z})$
PHIP1B	$\alpha f'(1/\overline{z})$
CONSET	unknown not in ϕ_z or ψ_z

α is a complex constant and $f(z)$ is a complex function (usually either $\phi_z(z)$, $\psi_z(z)$, $\phi(\zeta)$, or $\psi(\zeta)$).

Table A-4. SPECIALIZED EQUATION GENERATION ROUTINES

Subroutine	Use
FDCOL	Collocation of displacements and forces
STCOL	Collocation of stresses
STITCH	Equate forces and displacements at zone boundaries
COMBIN	Generation of equations for use in combined finite element - MMC analysis

Table A-5. RESULT COMPUTATION ROUTINES

Subroutine	Use
EVAL	Evaluate equation with solution
FDRES	Calculate forces and displacements with solution
STRRES	Calculate stresses with solutions
PRTCOF	Print coefficients solved for
PRTCON	Print unknown found which is not in functions
PRNTMP	Print mapping parameters
PRNTMT	Print material properties
PRNTSR	Print series information
PRTZON	Print all information in zone
K1K2	Calculation of stress intensities

APPENDIX B. SOURCE LISTINGS FOR MMCLIB

```
      FUNCTION BRANCH (TH,THBR)
C
C
C          FUNCTION TO FIND ANGLE TH+2.*N*PI SUCH THAT
C          THBR < ANGLE < THBR+2.*PI
C
C
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      TH1=TH
      THP=THBR+TWOPI
   10 IF (TH1.GE.THBR)  GO TO 20
      TH1=TH1+TWOPI
      GO TO 10
   20 IF (TH1.LE.THP)  GO TO 30
      TH1=TH1-TWOPI
      GO TO 20
   30 BRANCH=TH1
      RETURN
      END


      SUBROUTINE B4PHDP(XP,YP,IZ,CS2A,SN2A,IREF,A,B)
C
C
C          SUBROUTINE TO CALCULATE FUNCTION MULTIPLYING
C           SECOND DERIVATIVE OF PHI IN STRESS EQUATIONS
C
C
      CALL MAPPR(XP,YP,XPR,YPR,IZ)
      XBR=0.
      YBR=0.
      CALL MAPPER(XP,YP,XZBR,YZBR,IZ,1)
      YZBR=-YZBR
      GO TO (40,10,20),IREF
   10 CALL MAPB21(XP,YP,XBR,YBR,IZ)
      GO TO 30
   20 CALL MAPB12(XP,YP,XBR,YBR,IZ)
   30 YBR=-YBR
   40 XZBR=XBR-XZBR
      YZBR=YBR-YZBR
      XT=CS2A*XZBR-SN2A*YZBR
      YT=CS2A*YZBR+SN2A*XZBR
      XT1=XPR*XPR-YPR*YPR
      YT1=2.*XPR*YPR
      DENOM=1./(XT1*XT1+YT1*YT1)
      A=DENOM*(XT*XT1+YT*YT1)
      B=DENOM*(XT1*YT-YT1*XT)
      RETURN
      END


      SUBROUTINE B4PHIP (XP,YP,IZ,IOPT,C2,A,B)
C
C
C          SUBROUTINE TO CALCULATE FUNCTION MULTIPLYING COMPLEX
C           CONJUGATE OF DERIVATIVE OF PHI IN FORCE,DISPLACEMENT
C           EQUATIONS
C                IOPT
C                  1 - NO REFLECTION
C                  2 - X-AXIS REFLECTION
C                   3 - UNIT CIRCLE REFLECTION
C
C
```

```
      COMMON   /MAP/  IMAP(5,2),PARAM(5,2,5)
      CALL MAPPER (XP,YP,ZX,ZY,IZ,1)
      GO TO (40,10,20,60), IOPT
   10 CALL MAPB21 (XP,YP,ZXB,ZYB,IZ)
      GO TO 30
   20 CALL MAPB12 (XP,YP,ZXB,ZYB,IZ)
   30 ZX=ZX-ZXB
      ZY=ZY-ZYB
   40 XPR1=1.
      YPR1=0.
      IF (IMAP(IZ,1).EQ.0)  GO TO 50
      CALL MAPPBR (XP,YP,IZ,1,XPR1,YPR1)
   50 CALL MAPPRB (XP,YP,XPR2,YPR2,IZ)
      DENOM=C2/((XPR1*XPR1+YPR1*YPR1)*(XPR2*XPR2+YPR2*YPR2))
      ZXB=XPR1*XPR2-YPR1*YPR2
      ZYB=-XPR1*YPR2-XPR2*YPR1
      A=(ZX*ZXB-ZY*ZYB)*DENOM
      B=(ZY*ZXB+ZX*ZYB)*DENOM
      RETURN
   60 A=C2*ZX
      B=C2*ZY
      RETURN
      END


      SUBROUTINE B4PHP(XP,YP,CS2A,SN2A,IZ,IREF,A,B)
C
C
C          SUBROUTINE TO CALCULATE FUNCTION MULTIPLYING
C          DERIVATIVE OF PHI IN STRESS EQUATIONS
C
C
      COMMON   /MAP/  IMAP(5,2),PARAM(5,2,5)
      A=0.
      B=0.
      IF(IMAP(IZ,1).NE.0)GO TO 20
      IF(IREF.EQ.1)GO TO 80
      X2=1.
      Y2=0.
      IF(IREF.EQ.2)GO TO 10
      XT=XP*XP-YP*YP
      YT=2.*XP*YP
      DENOM=1./(XT*XT+YT*YT)
      X2=-XT*DENOM
      Y2= YT*DENOM
   10 A=X2*CS2A-Y2*SN2A
      B=X2*SN2A+Y2*CS2A
      GO TO 80
   20 CALL MAPPER(XP,YP,X2,Y2,IZ,1)
      Y2=-Y2
      CALL MAPPR(XP,YP,DERX,DERY,IZ)
      XT=DERX*(DERX*DERX-3.*DERY*DERY)
      YT=DERY*(3.*DERX*DERX-DERY*DERY)
      DENOM=1./(XT*XT+YT*YT)
      DX=XT*DENOM
      DY=-YT*DENOM
      CALL MAPPP1(XP,YP,IZ,1,X1,Y1)
      IF(IMAP(IZ,2).EQ.0)GO TO 30
      XT=X1
      YT=Y1
      CALL MAPPRP(XP,YP,XT1,YT1,IZ)
      X1=XT*XT1-YT*YT1
      Y1=XT*YT1+YT*XT1
      CALL MAPP1(XP,YP,IZ,1,XT1,YT1)
      XT=XT1*XT1-YT1*YT1
      YT=2.*XT1*YT1
      CALL MAP1(XP,YP,IZ,1,WX,WY)
```

```
      CALL MAPPP1(WX,WY,IZ,2,XT1,YT1)
      X1=X1+XT*XT1-YT*YT1
      Y1=Y1+XT*YT1+YT*XT1
   30 X3=0.
      Y3=0.
      GO TO (70,40,50),IREF
   40 CALL MAPB21(XP,YP,XT,YT,IZ)
      YT=-YT
      CALL MAPP21(XP,YP,XT1,YT1,IZ)
      GO TO 60
   50 CALL MAPP12(XP,YP,XT,YT,IZ)
      WX=XP*XP-YP*YP
      WY=2.*XP*YP
      DENOM=1./(WX*WX+WY*WY)
      XT1=-DENOM*(WX*XT+WY*YT)
      YT1=-DENOM*(WX*YT-WY*XT)
      CALL MAPB12(XP,YP,XT,YT,IZ)
      YT=-YT
   60 X2=X2-XT
      Y2=Y2-YT
      CALL MAPPR(XP,YP,XT,YT,IZ)
      X3=XT*XT1-YT*YT1
      Y3=XT*YT1+YT*XT1
   70 XT=X2*X1-Y2*Y1+X3
      YT=X2*Y1+Y2*X1+Y3
      XT1=DX*XT-DY*YT
      YT1=DX*YT+DY*XT
      A=CS2A*XT1-SN2A*YT1
      B=CS2A*YT1+SN2A*XT1
   80 RETURN
      END


      SUBROUTINE COMBIN (NFILE,IZ,IREF,IPRT,IERR)
C
C        ROUTINE TO WRITE EQUATIONS FOR
C        COMBINED FINITE ELEMENT - MMC
C        ANALYSIS
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON   /MAT/  G(5),ETAM(5)
      COMMON   /MMCFE/ X(150),Y(150),FDRHS(300),EQNS(120,176),
     *                KUV(120,2)
      DIMENSION RHS(1)
      REAL KUV
      EQUIVALENCE (RHS(1),FDRHS(1))
      DATA NEQ /100/
      ICNT=ICNT+1
      IF (IERR.NE.0.AND.ICNT.EQ.1)  GO TO 70
      DO 10  I=1,NEQ
      DO 10  J=1,176
   10 EQNS(I,J)=0.
      REWIND NFILE
      READ (NFILE) N1,N2,IDOFST
      IF (IPRT.EQ.2.AND.IERR.EQ.0)  PRINT 120,N1,N2
      NODST=(IDOFST+1)/2
      NPT=N2/2
      NODLST=NPT+NODST-1
      READ (NFILE) (X(I),Y(I),I=1,NPT)
      READ (NFILE) (RHS(I),I=1,N1)
      DO 50  N=1,NPT
      CALL MAPPER(XP,YP,X(N),Y(N),IZ,2)
      IF (IPRT.NE.2.OR.IERR.NE.0)  GO TO 20
      NODE=N+NODST-1
      PRINT 130, NODE,X(N),Y(N),XP,YP
   20 CALL ZERO (0,0,1,2,1,0)
      CALL PTCOL (XP,YP,IZ,2,IREF,1.,1,2,1,2)
```

```
      IF (IREF.EQ.4)  CALL MLTDSP (XP,YP,IZ)
      DO 30  J=1,2
   30 READ (NFILE) (KUV(I,J),I=1,N1)
      DO 40  I=1,N1
      DO 40  J=1,NUNK
   40 EQNS(I,J)=EQNS(I,J)+KUV(I,1)*EQUTOT(1,J)+KUV(I,2)*EQUTOT(2,J)
   50 CONTINUE
      TWOG=2.*G(IZ)
      DO 60  I=1,N1
   60 EQNS(I,NN)=RHS(I)*TWOG
      IF (IPRT.GT.0.AND.IERR.EQ.0)  PRINT 140, N1,NODST,NODLST
   70 N1HLF=N1/2
      DO 100  N=1,N1HLF
      IE2=2*N
      IE1=IE2-1
      DO 80  I=1,NN
      EQUTOT(1,I)=EQNS(IE1,I)
   80 EQUTOT(2,I)=EQNS(IE2,I)
      IF (IERR.NE.0)  GO TO 90
      CALL SAVE (1)
      GO TO 100
   90 CALL EVAL (1,2,2,XER,YER)
      PRINT 110, XER,YER
  100 CONTINUE
      RETURN
  110 FORMAT (1H ,2F10.4)
  120 FORMAT (////5H N1 =,I5,10X,4HN2 =,I5
     *         /////36H EQUATIONS FOR COMBINED METHOD TAKEN,
     1          28H FROM FOLLOWING NODAL POINTS//2X,4HNODE,11X,
     2          1HX,11X,1HY)
  130 FORMAT (1H ,I5,4E12.4)
  140 FORMAT (////I6,33H EQUATIONS GENERATED ON INTERFACE/
     *          9H AT NODES,I5,5H THRU,I5)
      END



      SUBROUTINE CONINF(ICONST,IRLIM)
C
C
C            SUBROUTINE TO SET UP FOR EXTRA UNKNOWNS SUCH AS FORCE CONSTAN
C                 ICONST - NUMBER OF EXTRA UNKNOWN
C                 IRLIM  - TWO DIGIT NUMBER
C                               01 -- UNKNOWN HAS ONLY IMAG PART
C                               10 -- UNKNOWN HAS ONLY REAL PART
C                               11 -- UNKNOWN HAS BOTH REAL, IMAG PARTS
C
C
      DIMENSION IIMRL(2)
      EQUIVALENCE (IRL,IIMRL(2)),(IIM,IIMRL(1))
      COMMON  /CONST/ ICON(5,2),ISTRT(5)
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      CALL DECODE(IRLIM,2,IIMRL,NDUM)
      IF(IRL.GT.0)IRL=1
      IF(IIM.GT.0)IIM=1
      NADD=IRL+IIM
      IF(NADD.LE.0)GO TO 10
      ICON(ICONST,1)=IRL
      ICON(ICONST,2)=IIM
      ISTRT(ICONST)=NUNK+1
      NUNK=NUNK+NADD
      NN=NUNK+1
   10 RETURN
      END
```

```
      SUBROUTINE CONSET(ICNST,XR1,XR2,XI1,XI2)
C
C
C          SUBROUTINE TO SET VALUES FOR COLLOCATION OF ADDED UNKNOWNS
C                ICNST - ADDED UNKNOWN NUMBER
C                XR1   - VALUE FOR REAL EQUATION, REAL PART OF UNKNOWN
C                XR2   - VALUE FOR REAL EQUATION, IMAG PART OF UNKNOWN
C                XI1   - VALUE FOR IMAG EQUATION, REAL PART OF UNKNOWN
C                XI2   - VALUE FOR IMAG EQUATION, IMAG PART OF UNKNOWN
C
C
      COMMON  /CONST/ ICON(5,2),ISTRT(5)
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      I1=ISTRT(ICNST)
      IF(ICON(ICNST,1).LE.0)GO TO 10
      EQUTOT(1,I1)=XR1
      EQUTOT(2,I1)=XI1
      I1=I1+1
   10 IF(ICON(ICNST,2).LE.0)GO TO 20
      EQUTOT(1,I1)=XR2
      EQUTOT(2,I1)=XI2
   20 RETURN
      END



      SUBROUTINE CRSMLT
C
C
C          SUBROUTINE TO READ EQUATIONS FROM TAPE AND PERFORM
C           SCALING, CROSS MULTIPLICATION
C
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON  /SOLN/  A(15401),IDIAG(176),PHI(176)
      REWIND IUNIT
      DO 40  I=1,NWRT
      READ (IUNIT) ((EQUTOT(J,K),K=1,NN),J=1,2)
      DO 10  J=1,2
      DO 10  K=1,NUNK
   10 EQUTOT(J,K)=EQUTOT(J,K)*SCALE(K)
      DO 30  L=1,2
      IND=0
      DO 20  J=1,NUNK
      DO 20  K=J,NUNK
      IND=IND+1
   20 A(IND)=A(IND)+EQUTOT(L,J)*EQUTOT(L,K)
      DO 30  J=1,NUNK
   30 PHI(J)=PHI(J)+EQUTOT(L,J)*EQUTOT(L,NN)
   40 CONTINUE
      RETURN
      END


      SUBROUTINE DECODE(IOPT,MAX,IDIGIT,NUM)
C         ROUTINE TO SEPARATE DIGITS FFROM
C         INTEGER CONTAINING STRING OF DIGITS
C
      DIMENSION IDIGIT(1)
      NUM=0
      ITSTR=1
      DO 10 I=1,MAX
   10 IDIGIT(I)=0
      DO 20 I=1,MAX
      IF(ITSTR.GT.IOPT)GO TO 30
```

```
      NUM=NUM+1
      IDEN=ITSTR
      ITSTR=10**I
   20 IDIGIT(I)=MOD(IOPT,ITSTR)/IDEN
   30 RETURN
      END


      SUBROUTINE DOF(NDOF)
C
C         ROUTINE TO INTERROGATE MMCLIB DATA BLOCKS
C         FOR NUMBER OF UNKNOWNS
C
      COMMON    /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                 SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      NDOF=NUNK
      RETURN
      END


      SUBROUTINE EVAL (IE1,IE2,IOPT,VAL1,VAL2)
C
C
C           SUBROUTINE TO EVALUATE EQUATIONS ONCE SOLUTION IS KNOWN
C
C
      COMMON    /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                 SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON    /SOLN/ A(15401),IDIAG(176),PHI(176)
      DIMENSION VALUE(2)
      DO 10  IE=IE1,IE2
      VALUE(IE)=0.
      IF (IOPT.EQ.2)  VALUE(IE)=-EQUTOT(IE,NN)
      DO 10  J=1,NUNK
   10 VALUE(IE)=VALUE(IE)+PHI(J)*EQUTOT(IE,J)
      VAL1=VALUE(1)
      VAL2=VALUE(2)
      RETURN
      END


      SUBROUTINE FDRES (IZ,NPT,IOPT,IREF,X,Y,XP,YP)
C
C
C               SUBROUTINE TO PRINT FORCES, DISPLACEMENTS FROM SOLUTION
C
C                     IOPT      - 1 FOR FORCES
C                                 2 FOR DISPLACEMENTS
C                     IREF      - 1 FOR NO REFLECTIONS IN ZONE
C                                 2 FOR X-AXIS REFLECTION
C                                 3 FOR UNIT CIRCLE REFLECTION
C
C
      DIMENSION X(1), Y(1), XP(1), YP(1)
      COMMON    /MAT/  G(5),ETAM(5)
      A=1.
      B=0.
      GO TO (10,20), IOPT
   10 PRINT 50
      GO TO 30
   20 PRINT 60
   30 DO 40  I=1,NPT
      CALL ZERO (0,0,1,2,1,0)
      CALL PTCOL (XP(I),YP(I),IZ,IOPT,IREF,1.,1,2,1,2)
```

```
      IF (IREF.EQ.4)  CALL MAPPR (XP(I),YP(I),A,B,IZ)
      FACT=1./(A*A+B*B)
      IF (IOPT.EQ.2)  FACT=FACT/(2.*G(IZ))
      CALL EVAL (1,2,1,V1,V2)
      VALX=(V1*A-V2*B)*FACT
      VALY=(V1*B+V2*A)*FACT
   40 PRINT 70, VALX,VALY,X(I),Y(I),XP(I),YP(I)
      RETURN
   50 FORMAT (21HOEVALUATION OF FORCES / 9X,2HF1,11X,2HF2,12X,1HX,12X,1H
     1Y        8X,8HX(PARAM),5X,8HY(PARAM) )
   60 FORMAT (28HOEVALUATION OF DISPLACEMENTS / 10X,1HU,12X,1HV,12X,1HX,
     1         12X,1HY,8X,8HX(PARAM),5X,8HY(PARAM) )
   70 FORMAT (1X,6E13.5)
      END


      SUBROUTINE FDCOL (XP,YP,N,IZ,IOPT,IREF,RHS1,RHS2,IERR)
C
C
C           SUBROUTINE TO COMPUTE FORCE, DISPLACEMENT EQUATIONS
C                 MEANING OF IREF
C                   1 - NO REFLECTIONS
C                   2 - X-AXIS REFLECTION
C                   3 - UNIT CIRCLE REFLECTION
C
C
      DIMENSION XP(1), YP(1), RHS1(1), RHS2(1)
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      PRINT 40
      A=1.
      B=0.
      DO 20  I=1,N
      CALL ZERO (0,0,1,2,1,0)
      CALL PTCOL (XP(I),YP(I),IZ,IOPT,IREF,1.,1,2,1,2)
      IF(IREF.EQ.4)CALL MAPPR(XP(I),YP(I),A,B,IZ)
      EQUTOT(1,NN)=A*RHS1(I)+B*RHS2(I)
      EQUTOT(2,NN)=A*RHS2(I)-B*RHS1(I)
      IF (IERR.EQ.1)  GO TO 10
      CALL SAVE (1)
      GO TO 20
   10 CALL EVAL (1,2,2,XER,YER)
      PRINT 30, XER,YER,XP(I),YP(I)
   20 CONTINUE
      RETURN
   30 FORMAT (1H ,2F10.4,2E10.4)
   40 FORMAT(1H )
      END


      SUBROUTINE FNDMAT
C
C
C           SUBROUTINE TO SCALE, CROSS-MULTIPLY, AND SOLVE
C            FOR COEFFICIENTS
C
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON   /SOLN/ A(15401),IDIAG(176),PHI(176)
      CALL SLVINT
      CALL SCLFIX
      CALL CRSMLT
      CALL SOLVE
      DO 10  I=1,NUNK
   10 PHI(I)=PHI(I)*SCALE(I)
      RETURN
      END
```

```
      SUBROUTINE INIT
C
C
C          SUBROUTINE TO INITIALIZE ALL COMMON VARIABLES
C
C
      COMMON  /CONST/  ICON(5,2),ISTRT(5)
      COMMON    /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                 SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON    /MAP/  IMAP(5,2),PARAM(5,2,5)
      COMMON    /MAT/  G(5),ETAM(5)
      COMMON  /PIFAC/  PI,HLFPI,TWOPI
      COMMON /POWERS/  R(100),TH(100),ISHFT
      COMMON /SERIES/  ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                 NPOS(10),IZRO(10),IODRL(10),
     *                 IODIM(10),IEVRL(10),IEVIM(10)
      COMMON   /SOLN/  A(15401),IDIAG(176),PHI(176)
      PI=3.14159265
      HLFPI=1.57079633
      TWOPI=6.28318531
      IUNIT=7
      ISHFT=0
      DO 10  I=1,10
      IBEGI(I)=0
      IENDI(I)=0
      NNEG(I)=0
      NPOS(I)=0
      IZRO(I)=0
      IODRL(I)=0
      IODIM(I)=0
      IEVRL(I)=0
      IEVIM(I)=0
      XNOT(I)=0.
   10 YNOT(I)=0.
      DO 20  I=1,5
      ISTRT(I)=0
      G(I)=0.
      ETAM(I)=0.
      DO 20  J=1,2
      IMAP(I,J)=0
      ICON(I,J)=0
      DO 20  K=1,5
   20 PARAM(I,J,K)=0.
      REWIND IUNIT
      IZONES=0
      NWRT=0
      NUNK=0
      NUNKS=0
      NN=0
      DO 30  I=1,175
   30 SCALE(I)=0.
      RETURN
      END


      SUBROUTINE K1K2(XP,YP,A,B,K1,K2,IZ)
C
C        ROUTINE TO COMPUTE STRESS
C        INTENSITY FACTORS
C
      REAL K1,K2
      CALL ZERO(1,2,1,2,3,0)
      CALL PHIPB(XP,YP,A,-B,IZ)
      CALL PLACE(IZ,1,2,1,2,1.)
      CALL EVAL(1,2,1,K1,K2)
      RETURN
      END
```

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS
USING THE MODIFIED MAPPING-COLLOCATION METHOD -
Peter G. Tracy

AD____________

UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer programs
Stress analysis
Complex variables

Technical Report AMMRC TR 79-13, March 1979, 53 pp -
illus-tables, D/A Project 1T161101A91A
AMCMS Code 611101.91A0011

A computer code for the solution of a wide variety of plane isotropic stress problems using the modified mapping-collocation method is presented. The mathematical approach, capabilities, and structure of the code are examined. Finally, a sample problem is solved to illustrate the ease with which stress solutions can be obtained utilizing the code.

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS
USING THE MODIFIED MAPPING-COLLOCATION METHOD -
Peter G. Tracy

AD____________

UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer programs
Stress analysis
Complex variables

Technical Report AMMRC TR 79-13, March 1979, 53 pp -
illus-tables, D/A Project 1T161101A91A
AMCMS Code 611101.91A0011

A computer code for the solution of a wide variety of plane isotropic stress problems using the modified mapping-collocation method is presented. The mathematical approach, capabilities, and structure of the code are examined. Finally, a sample problem is solved to illustrate the ease with which stress solutions can be obtained utilizing the code.

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS
USING THE MODIFIED MAPPING-COLLOCATION METHOD -
Peter G. Tracy

AD____________

UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer programs
Stress analysis
Complex variables

Technical Report AMMRC TR 79-13, March 1979, 53 pp -
illus-tables, D/A Project 1T161101A91A
AMCMS Code 611101.91A0011

A computer code for the solution of a wide variety of plane isotropic stress problems using the modified mapping-collocation method is presented. The mathematical approach, capabilities, and structure of the code are examined. Finally, a sample problem is solved to illustrate the ease with which stress solutions can be obtained utilizing the code.

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS
USING THE MODIFIED MAPPING-COLLOCATION METHOD -
Peter G. Tracy

AD____________

UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer programs
Stress analysis
Complex variables

Technical Report AMMRC TR 79-13, March 1979, 53 pp -
illus-tables, D/A Project 1T161101A91A
AMCMS Code 611101.91A0011

A computer code for the solution of a wide variety of plane isotropic stress problems using the modified mapping-collocation method is presented. The mathematical approach, capabilities, and structure of the code are examined. Finally, a sample problem is solved to illustrate the ease with which stress solutions can be obtained utilizing the code.

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS
USING THE MODIFIED MAPPING-COLLOCATION METHOD -
Peter G. Tracy

AD________________

UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer programs
Stress analysis
Complex variables

Technical Report AMMRC TR 79-13, March 1979, 53 pp -
illus-tables, D/A Project 1T161101A91A
AMCMS Code 611101.91A0011

A computer code for the solution of a wide variety of plane isotropic stress problems using the modified mapping-collocation method is presented. The mathematical approach, capabilities, and structure of the code are examined. Finally, a sample problem is solved to illustrate the ease with which stress solutions can be obtained utilizing the code.

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS
USING THE MODIFIED MAPPING-COLLOCATION METHOD -
Peter G. Tracy

AD________________

UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer programs
Stress analysis
Complex variables

Technical Report AMMRC TR 79-13, March 1979, 53 pp -
illus-tables, D/A Project 1T161101A91A
AMCMS Code 611101.91A0011

A computer code for the solution of a wide variety of plane isotropic stress problems using the modified mapping-collocation method :s presented. The mathematical approach, capabilities, and structure of the code are examined. Finally, a sample problem is solved to illustrate the ease with which stress solutions can be obtained utilizing the code.

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS
USING THE MODIFIED MAPPING-COLLOCATION METHOD -
Peter G. Tracy

AD________________

UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer programs
Stress analysis
Complex variables

Technical Report AMMRC TR 79-13, March 1979, 53 pp -
illus-tables, D/A Project 1T161101A91A
AMCMS Code 611101.91A0011

A computer code for the solution of a wide variety of plane isotropic stress problems using the modified mapping-collocation method is presented. The mathematical approach, capabilities, and structure of the code are examined. Finally, a sample problem is solved to illustrate the ease with which stress solutions can be obtained utilizing the code.

Army Materials and Mechanics Research Center
Watertown, Massachusetts 02172
MMCLIB: A COMPUTER CODE FOR STRESS ANALYSIS
USING THE MODIFIED MAPPING-COLLOCATION METHOD -
Peter G. Tracy

AD________________

UNCLASSIFIED
UNLIMITED DISTRIBUTION

Key Words

Computer programs
Stress analysis
Complex variables

Technical Report AMMRC TR 79-13, March 1979, 53 pp -
illus-tables, D/A Project 1T161101A91A
AMCMS Code 611101.91A0011

A computer code for the solution of a wide variety of plane isotropic stress problems using the modified mapping-collocation method is presented. The mathematical approach, capabilities, and structure of the code are examined. Finally, a sample problem is solved to illustrate the ease with which stress solutions can be obtained utilizing the code.

```
      SUBROUTINE MAP (XP,YP,P1,P2,P3,P4,P5,M,X,Y)
C
C
C          SUBROUTINE TO CALCULATE MAPPING FUNCTIONS
C
      GO TO (10,20), M
C
C          MAPPING #1
C
   10 CSPHI=COS(P3)
      SNPHI=SIN(P3)
      RSQ=XP*XP+YP*YP
      RSQS=RSQ+1.
      RSQD=RSQ-1.
      SUM1=(P1*RSQS+P2*RSQD)*XP
      SUM2=(P1*RSQD+P2*RSQS)*YP
      DENOM=1./(2.*RSQ)
      X=P4+(CSPHI*SUM1-SNPHI*SUM2)*DENOM
      Y=P5+(SNPHI*SUM1+CSPHI*SUM2)*DENOM
      GO TO 30
C
C          MAPPING #2
C
   20 CSPHI=COS(P3)
      SNPHI=SIN(P3)
      SNHY=SINH(YP)
      CSHY=COSH(YP)
      SUM1=SIN(XP)*(P2*SNHY+P1*CSHY)
      SUM2=COS(XP)*(P2*CSHY+P1*SNHY)
      X=P4+CSPHI*SUM1-SNPHI*SUM2
      Y=P5+SNPHI*SUM1+CSPHI*SUM2
   30 RETURN
      END


      SUBROUTINE MAPB (XP,YP,IZ,I12,X,Y)
      CALL PARA (IZ,I12,M,P1,P2,P3,P4,P5)
      CALL MAP (XP,-YP,P1,P2,P3,P4,P5,M,X,Y)
      RETURN
      END


      SUBROUTINE MAPBAR (XP,YP,IZ,I12,X,Y)
      CALL PARA (IZ,I12,M,P1,P2,P3,P4,P5)
      CALL MAP (XP,YP,P1,P2,P3,P4,P5,M,X,Y)
      Y=-Y
      RETURN
      END


      SUBROUTINE MAPBR (XP,YP,IZ,I12,X,Y)
      CALL PARA (IZ,I12,M,P1,P2,P3,P4,P5)
      GO TO (10,20), M
   10 BB=P2
      GO TO 30
   20 BB=-P2
   30 CALL MAP (XP,YP,P1,BB,P3,P4,-P5,M,X,Y)
      RETURN
      END


      SUBROUTINE MAPBR1 (XP,YP,IZ,I12,X,Y)
      CALL MAPB1 (XP,YP,IZ,I12,X,Y)
      Y=-Y
      RETURN
      END
```

```
      SUBROUTINE MAPB1 (XP,YP,IZ,I12,X,Y)
      CALL PARA (IZ,I12,M,P1,P2,P3,P4,P5)
      R1=1./(XP*XP+YP*YP)
      XT=XP*R1
      YT=YP*R1
      CALL MAP (XT,YT,P1,P2,P3,P4,P5,M,X,Y)
      RETURN
      END


      SUBROUTINE MAPB12 (XP,YP,X,Y,IZ)
      COMMON    /MAP/  IMAP(5,2),PARAM(5,2,5)
      R1=1./(XP*XP+YP*YP)
      X=XP*R1
      Y=YP*R1
      IF (IMAP(IZ,1).EQ.0)  RETURN
      CALL MAPB1 (XP,YP,IZ,1,X,Y)
      IF (IMAP(IZ,2).EQ.0)  RETURN
      XX=X
      YY=Y
      CALL MAP1 (XX,YY,IZ,2,X,Y)
      RETURN
      END


      SUBROUTINE MAPB21 (XP,YP,X,Y,IZ)
      COMMON    /MAP/  IMAP(5,2),PARAM(5,2,5)
      X=XP
      Y=-YP
      IF (IMAP(IZ,1).EQ.0)  RETURN
      CALL MAPB (XP,YP,IZ,1,X,Y)
      IF (IMAP(IZ,2).EQ.0)  RETURN
      XX=X
      YY=Y
      CALL MAP1 (XX,YY,IZ,2,X,Y)
      RETURN
      END


      SUBROUTINE MAPINF (IZ,M1,P11,P12,P13,P14,P15,M2,P21,P22,
     1                   P23,P24,P25)
C
C
C         SUBROUTINE TO SET MAPPING FUNCTION VARIABLES
C
C
      COMMON    /MAP/  IMAP(5,2),PARAM(5,2,5)
      DATA ANGCON /.17453292E-1/
      IF (M1.LE.0)  RETURN
      PARAM(IZ,1,1)=P11
      PARAM(IZ,1,2)=P12
      PARAM(IZ,1,3)=P13*ANGCON
      PARAM(IZ,1,4)=P14
      PARAM(IZ,1,5)=P15
      IMAP(IZ,1)=M1
      IF (M2.LE.0)  RETURN
      PARAM(IZ,2,1)=P21
      PARAM(IZ,2,2)=P22
      PARAM(IZ,2,3)=P23*ANGCON
      PARAM(IZ,2,4)=P24
      PARAM(IZ,2,5)=P25
      IMAP(IZ,2)=M2
      RETURN
      END
```

```
      SUBROUTINE MAPP (XP,YP,P1,P2,P3,P4,P5,M,XD,YD)
C
C        ROUTINE TO COMPUTE FIRST DERIVATIVE
C        OF MAPPING FUNCTION
C
      COMMON   /MAP/  IMAP(5,2),PARAM(5,2,5)
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      GO TO (10,20), M
C
C        MAPPING #1
C
   10 BMA=P2-P1
      R4=XP*XP+YP*YP
      R4=R4*R4
      R41=1./R4
      FACT1=.5*(BMA*(XP*XP-YP*YP)+R4*(P1+P2))
      FACT2=BMA*XP*YP
      CS=COS(P3)
      SN=SIN(P3)
      XD=R41*(FACT1*CS+FACT2*SN)
      YD=R41*(FACT1*SN-FACT2*CS)
      RETURN
   20 CALL MAP (XP,YP,P2,P1,P3-HLFPI,0.,0.,M,XD,YD)
      RETURN
C
      END


      SUBROUTINE MAPINV (X,Y,P1,P2,P3,P4,P5,M,XP,YP)
C
C
C           SUBROUTINE TO CALCULATE MAPPING FUNCTION INVERSES
C
C
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      GO TO (10,20), M
C
C        MAPPING #1
C
   10 CALL QUADF (P1,P2,P3,P4,P5,X,Y,1,XP,YP)
      RETURN
C
C        MAPPING #2
C
   20 IMAPR=2
   30 CALL QUADF (P2,-P1,P3+HLFPI,P4,P5,X,Y,IMAPR,XX,YY)
      XP=XX
      YP=YY
      IF (P1.EQ.P2)  GO TO 50
      XP=ATAN2(YY,XX)
      YP=-ALOG(SQRT(XX*XX+YY*YY))
      IF (YP.GT.-.0001)  GO TO 50
      IF (IMAPR.NE.-2)  GO TO 40
      WRITE (6,60) M,X,Y
      STOP
   40 IMAPR=-2
      GO TO 30
   50 RETURN
   60 FORMAT (17H ERROR IN INVERSE ,I3,2E12.5)
      END

      SUBROUTINE MAPPB1 (XP,YP,IZ,I12,XD,YD)
      CALL PARA (IZ,I12,M,P1,P2,P3,P4,P5)
      GO TO (10,20), M
   10 CALL MAPP ( XP, YP,P1,P2,-P3,P4,P5,M,XD,YD)
      RETURN
   20 CALL MAPP (-XP,-YP,P1,P2,-P3,P4,P5,M,XD,YD)
      RETURN
      END
```

```
      SUBROUTINE MAPPBR (XP,YP,IZ,I12,XD,YD)
      CALL PARA (IZ,I12,M,P1,P2,P3,P4,P5)
      CALL MAPP (XP,YP,P1,P2,P3,P4,P5,M,XD,YD)
      YD=-YD
      RETURN
      END


      SUBROUTINE MAPPB2 (XP,YP,IZ,I12,XD,YD)
      RR=1./SQRT(XP*XP+YP*YP)
      XX=XP*RR
      YY=-YP*RR
      CALL MAPPB1 (XX,YY,IZ,I12,XD,YD)
      RETURN
      END


      SUBROUTINE MAPPDP(XP,YP,XD,YD,IZ)
      COMMON   /MAP/  IMAP(5,2),PARAM(5,2,5)
      XD=1.
      YD=0.
      IF(IMAP(IZ,1).EQ.0.AND.IMAP(IZ,2).EQ.0)RETURN
      CALL MAPPP1(XP,YP,IZ,1,ZMX2,ZMY2)
      IF(IMAP(IZ,2).NE.0)GO TO 10
      XD=ZMX2
      YD=ZMY2
      RETURN
   10 CALL MAPP1 (XP,YP,IZ,1,ZMX1,ZMY1)
      ZMX1SQ=ZMX1*ZMX1-ZMY1*ZMY1
      ZMY1SQ=2.*ZMX1*ZMY1
      CALL MAPB(XP,-YP,IZ,WXP,WY)
      CALL MAPP1 (WX,WY,IZ,2,WMX1,WMY1)
      CALL MAPPP1(WX,WY,IZ,2,WMX2,WMY2)
      XD=WMX2*ZMX1SQ-WMY2*ZMY1SQ+WMX1*ZMX2-WMY1*ZMY2
      YD=WMX2*ZMY1SQ+WMY2*ZMX1SQ+WMX1*ZMY2+WMY1*ZMX2
      RETURN
      END


      SUBROUTINE MAPPP1 (XP,YP,IZ,I12,XD2,YD2)
      CALL PARA (IZ,I12,M,P1,P2,P3,P4,P5)
      CALL MAPPP (XP,YP,P1,P2,P3,P4,P5,M,XD2,YD2)
      RETURN
      END


      SUBROUTINE MAPPR(XP,YP,XD,YD,IZ)
      COMMON   /MAP/  IMAP(5,2),PARAM(5,2,5)
      XD=1.
      YD=0.
      IF(IMAP(IZ,1).EQ.0)GO TO 10
      CALL MAPP1(XP,YP,IZ,1,XP1,YP1)
      CALL MAPPRP(XP,YP,XP2,YP2,IZ)
      XD=XP1*XP2-YP1*YP2
      YD=XP1*YP2+XP2*YP1
   10 RETURN
      END
```

```
      SUBROUTINE MAPPER (XP,YP,X,Y,IZ,INV)
C
C
C          SUBROUTINE TO CALCULATE MAPPING FUNCTIONS IN ZONES
C
C
      COMMON   /MAP/  IMAP(5,2),PARAM(5,2,5)
      XX=XP
      YY=YP
      XXM=X
      YYM=Y
      IMP=2
      IF (IMAP(IZ,1).GT.0.AND. IMAP(IZ,2).GT.0)  GO TO 30
      IMP=1
      IF (IMAP(IZ,1).GT.0)  GO TO 30
      GO TO (10,20), INV
   10 X=XP
      Y=YP
      RETURN
   20 XP=X
      YP=Y
      RETURN
   30 DO 60  I=1,IMP
      II=I
      IF (INV.EQ.2.AND.IMP.EQ.2)  II=3-I
      CALL PARA (IZ,II,M,P1,P2,P3,P4,P5)
      GO TO (40,50), INV
   40 CALL MAP (XX,YY,P1,P2,P3,P4,P5,M,X,Y)
      XX=X
      YY=Y
      GO TO 60
   50 CALL MAPINV (XXM,YYM,P1,P2,P3,P4,P5,M,XP,YP)
      XXM=XP
      YYM=YP
   60 CONTINUE
      RETURN
      END



      SUBROUTINE MAPP1 (XP,YP,IZ,I12,XD,YD)
      CALL PARA (IZ,I12,M,P1,P2,P3,P4,P5)
      CALL MAPP (XP,YP,P1,P2,P3,P4,P5,M,XD,YD)
      RETURN
      END


      SUBROUTINE MAPPP(XP,YP,P1,P2,P3,P4,P5,M,XD2,YD2)
C
C        ROUTINE TO COMPUTE SECOND DERIVATIVE
C        OF MAPPING FUNCTION
C
      GO TO (10,20), M
C
C        MAPPING #1
C
   10 R6=XP*XP+YP*YP
      R6=(P1-P2)/(R6*R6*R6)
      FACT1=XP*XP*XP-3.*XP*YP*YP
      FACT2=YP*YP*YP-3.*YP*XP*XP
      CS=COS(P3)
      SN=SIN(P3)
      XD2=R6*(FACT1*CS-FACT2*SN)
      YD2=R6*(FACT1*SN+FACT2*CS)
      RETURN
C
C        MAPPING #2
```

```
C
   20 CALL MAP (XP,YP,P1,P2,P3,0.,0.,M,XD2,YD2)
      XD2=-XD2
      YD2=-YD2
      RETURN
      END


      SUBROUTINE MAPPR8 (XP,YP,XD,YD,IZ)
      CALL MAPPRP (XP,YP,XD,YD,IZ)
      YD=-YD
      RETURN
      END


      SUBROUTINE MAPPRP (XP,YP,XD,YD,IZ)
      COMMON   /MAP/  IMAP(5,2),PARAM(5,2,5)
      XD=1.
      YD=0.
      IF (IMAP(IZ,1).EQ.0.OR.IMAP(IZ,2).EQ.0)  RETURN
      CALL MAP1  (XP,YP,IZ,1,XX,YY)
      CALL MAPP1 (XX,YY,IZ,2,XD,YD)
      RETURN
      END


      SUBROUTINE MAPP12(XP,YP,XD,YD,IZ)
      DENOM=1./(XP*XP+YP*YP)
      X1=XP*DENOM
      Y1=YP*DENOM
      CALL MAPPR(X1,Y1,XP,YD,IZ)
      YD=-YD
      RETURN
      END


      SUBROUTINE MAPP21(XP,YP,XD,YD,IZ)
      CALL MAPPR(XP,-YP,XD,YD,IZ)
      YD=-YD
      RETURN
      END


      SUBROUTINE MAP1 (XP,YP,IZ,I12,X,Y)
      CALL PARA (IZ,I12,M,P1,P2,P3,P4,P5)
      CALL MAP (XP,YP,P1,P2,P3,P4,P5,M,X,Y)
      RETURN
      END


      SUBROUTINE MLTDSP(XP,YP,IZ)
      CALL MAPPR(XP,YP,A,B,IZ)
      AA=A/(A*A+B*B)
      BB=B/(A*A+B*B)
      CALL ROTZON(3,IZ,AA,-BB,BB,AA)
      RETURN
      END
```

```
      SUBROUTINE NWRTFL(N)
C
C           SUBROUTINE TO ALTER VARIABLE CONTAINING NUMBER
C             OF EQUATIONS WRITTEN ON CURRENT SCRATCH FILE
C
      COMMON    /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                 SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      NWRT=N
      RETURN
      END


      FUNCTION NWRTSV(IDUM)
C
C             FUNCTION TO SAVE NUMBER OF EQUATIONS WRITTEN
C              ON CURRENT SCRATCH UNIT
C
      COMMON    /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                 SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      NWRTSV=NWRT
      RETURN
      END


      SUBROUTINE PARA (IZ,I12,M,P1,P2,P3,P4,P5)
C
C
C           SUBROUTINE TO FIND MAPPING FUNCTION PARAMETERS IN ZONE
C
C
      COMMON    /MAP/   IMAP(5,2),PARAM(5,2,5)
      M =IMAP (IZ,I12)
      P1=PARAM(IZ,I12,1)
      P2=PARAM(IZ,I12,2)
      P3=PARAM(IZ,I12,3)
      P4=PARAM(IZ,I12,4)
      P5=PARAM(IZ,I12,5)
      RETURN
      END


      SUBROUTINE PHI (XP,YP,A,B,IZ)
C
C        ROUTINE TO COLLOCATE A FUNCTION IN A ZONE
C
      CALL ZETN (XP,YP,IZ)
      CALL SERFIL (IZ,1.,1.,A,B)
      RETURN
      END


      SUBROUTINE PHIB (XP,YP,A,B,IZ)
      CALL PHI (XP,-YP,A,B,IZ)
      RETURN
      END


      SUBROUTINE PHIBP (XP,YP,A,B,IZ)
      CALL ZETN1 (XP,YP,IZ)
      CALL SERFIL (IZ,1.,-1.,A,B)
      RETURN
      END
```

```
      SUBROUTINE PHIBP1(XP,YP,A,B,IZ)
      CALL PHIP(XP,-YP,A,B,IZ)
      RETURN
      END


      SUBROUTINE PHIBR(XP,YP,A,B,IZ)
      CALL ZETN(XP,YP,IZ)
      CALL SERFIL(IZ,-1.,-1.,A,B)
      RETURN
      END


      SUBROUTINE PHIB1 (XP,YP,A,B,IZ)
      R1=1./(XP*XP+YP*YP)
      CALL PHI (XP*R1,YP*R1,A,B,IZ)
      RETURN
      END


      SUBROUTINE PHIB1P(XP,YP,A,B,IZ)
      RR=1./(XP*XP+YP*YP)
      CALL PHIBP (XP*RR,-YP*RR,A,B,IZ)
      RETURN
      END


      SUBROUTINE PHIP(XP,YP,A,B,IZ)
C
C        ROUTINE TO COLLOCATE DERIVATIVE
C        OF A FUNCTION
C
      CALL ZETN1(XP,YP,IZ)
      CALL SERFIL(IZ,1.,1.,A,B)
      RETURN
      END


      SUBROUTINE PHIPB(XP,YP,A,B,IZ)
      CALL ZETN1(XP,YP,IZ)
      CALL SERFIL(IZ,-1.,-1.,A,B)
      RETURN
      END


      SUBROUTINE PHIPP (XP,YP,A,B,IZ)
C
C        ROUTINE TO COLLOCATE SECOND
C        DERIVATIVE OF A FUNCTION IN A ZONE
C
      CALL ZETN2 (XP,YP,IZ)
      CALL SERFIL (IZ,1.,1.,A,B)
      RETURN
      END
```

```
      SUBROUTINE PHIPPB(XP,YP,A,B,IZ)
      CALL ZETN2(XP,YP,IZ)
      CALL SERFIL(IZ,-1.,-1.,A,B)
      RETURN
      END



      SUBROUTINE PHIP1B(XP,YP,A,B,IZ)
      RR=1./(XP*XP+YP*YP)
      CALL PHIP(XP*RR,YP*RR,A,B,IZ)
      RETURN
      END


      SUBROUTINE PLACE (IZ,IE1,IE2,IT1,IT2,WT)
C
C
C           SUBROUTINE TO PUT COEFFICIENTS FROM ZONE EQUATION
C             INTO MASTER EQUATION
C
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      NTRMS=IENDI(IZ)-IBEGI(IZ)+1
      ITOT=IE2-IE1+1
      DO 10  I=1,ITOT
      IE=I+IE1-1
      IT=I+IT1-1
      DO 10  J=1,NTRMS
      J1=J+IBEGI(IZ)-1
   10 EQUTOT(IT,J1)=EQUTOT(IT,J1)+EQU(IE,J)*WT
      RETURN
      END



      SUBROUTINE POINTF (IZ,X1,Y1,X2,Y2,NPT,NBEG,NNEND,ICTYP,GPAR,
     2                   IMAP,INV,X,Y,XPAR,YPAR)
C
C
C            SUBROUTINE TO FIND POINTS AT WHICH TO COLLOCATE
C
C                 IZ            - ZONE NUMBER
C                 X1 + I Y1     - BEGINNING POINT
C                 X2 + I Y2     - ENDING POINT
C                 NBEG,NNEND    - NUMBER OF POINTS TO LEAVE OUT AT
C                                 BEGINNING,END
C                 ICTYP         - 1 -- STRAIGHT LINE
C                                 2 -- CIRCULAR ARC
C                                 3 -- ELLIPTICAL ARC
C                 IMAP          - MAP COORDINATES UNLESS 0
C                 INV           - 1 -- MAPPING FROM PARAMETER
C                               - 2 -- MAPPING TO PARAMETER
C                 GPAR(1)       - AXIS IN X-DIRECTION(ICTYP=2,3)
C                 GPAR(2)       - AXIS IN Y-DIRECTION(ICTYP=2,3)
C                 GPAR(3)       - ANGLE OF ROTATION (ICTYP=3)
C                 GPAR(4)       - X0 (ICTYP=2,3)
C                 GPAR(5)       - Y0 (ICTYP=2,3)
C
C
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      DIMENSION X(1), Y(1), XPAR(1), YPAR(1),GPAR(1)
      DATA ICLOCK /1/
      GPAR3R=GPAR(3)*.17453292E-1
      XN=NPT-1
```

```
      NSTP=NPT-NNEND
      GO TO (10,20,30), ICTYP
   10 FACT1=(X2-X1)/XN
      FACT2=(Y2-Y1)/XN
      GO TO 70
   20 XX=X1-GPAR(4)
      YY=Y1-GPAR(5)
      TH1=TANGO(YY,XX)
      XX=X2-GPAR(4)
      YY=Y2-GPAR(5)
      TH2=TANGO(YY,XX)
      GO TO 40
   30 CALL QUADF (GPAR(1),GPAR(2),GPAR3R,GPAR(4),GPAR(5),X1,Y1,1,XX,YY)
      TH1=TANGO(YY,XX)
      CALL QUADF (GPAR(1),GPAR(2),GPAR3R,GPAR(4),GPAR(5),X2,Y2,1,XX,YY)
      TH2=TANGO(YY,XX)
   40 IF (ICLOCK.EQ.-1)  GO TO 50
      IF (TH2.LT.TH1)  TH2=TH2+TWOPI
      GO TO 60
   50 IF (TH2.GT.TH1)  TH2=TH2-TWOPI
   60 TH0=.5*(TH1+TH2)+PI
      TH0=BRANCH(TH0,-PI)
      TH1=BRANCH(TH1,TH0)
      TH2=BRANCH(TH2,TH0)
      THETA=(TH2-TH1)/XN
   70 DO 110  I=1,NPT
      IF (I.LE.NBEG.OR.I.GT.NSTP)  GO TO 110
      XNN=I-1
      I1=I-NBEG
      GO TO (80,90,90), ICTYP
   80 X(I1)=X1+XNN*FACT1
      Y(I1)=Y1+XNN*FACT2
      GO TO 110
   90 ANG=TH1+XNN*THETA
      IF (ICTYP.EQ.3)  GO TO 100
      X(I1)=GPAR(4)+GPAR(1)*COS(ANG)
      Y(I1)=GPAR(5)+GPAR(1)*SIN(ANG)
      GO TO 110
  100 XX=COS(ANG)
      YY=SIN(ANG)
      CALL MAP (XX,YY,GPAR(1),GPAR(2),GPAR3R,GPAR(4),GPAR(5),
     *          1,X(I1),Y(I1))
  110 CONTINUE
      NPT=NPT-NBEG-NNEND
      IF (INV.EQ.2)  GO TO 130
      DO 120  I=1,NPT
      XPAR(I)=X(I)
  120 YPAR(I)=Y(I)
  130 IF (IMAP.EQ.0)  RETURN
      DO 140  I=1,NPT
  140 CALL MAPPER (XPAR(I),YPAR(I),X(I),Y(I),IZ,INV)
      RETURN
      END


      SUBROUTINE POWSER(XP,YP,A,B,IZ)
C
C         ROUTINE TO COMPUTE TERMS
C         FOR POWER SERIES
C
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      COMMON /POWERS/ R(100),TH(100),ISHFT
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      XX=XP-XNOT(IZ)
      YY=YP-YNOT(IZ)
      IPOW=-NNEG(IZ)
```

```
      NTOT1=NNEG(IZ)+NPOS(IZ)+1
      ISTRT=ISHFT+1
      R1=SQRT(XX*XX+YY*YY)
      TH1=ATAN2(YY,XX)
      XIPOW=IPOW
      R(ISTRT)=SQRT(A*A+B*B)*(R1**IPOW)
      TH(ISTRT)=BRANCH(ATAN2(B,A)+TH1*XIPOW,-PI)
      DO 10 I=2,NTOT1
      I1=ISHFT+I
      R(I1)=R(I1-1)*R1
   10 TH(I1)=BRANCH(TH(I1-1)+TH1,-PI)
      RETURN
      END


      SUBROUTINE PRNTMP (IZ)
C
C         ROUTINE TO PRINT MAPPING
C         INFORMATION IN A ZONE
C
      COMMON   /MAP/  IMAP(5,2),PARAM(5,2,5)
      IMP=1
      PRINT 20, IZ
      IF (IMAP(IZ,1).GT.0)  GO TO 10
      PRINT 30
      RETURN
   10 PRINT 40, IMP,IMAP(IZ,1),(PARAM(IZ,1,I),I=1,5)
      IF (IMAP(IZ,2).LE.0)  RETURN
      IMP=2
      PRINT 40, IMP,IMAP(IZ,2),(PARAM(IZ,2,I),I=1,5)
      RETURN
   20 FORMAT (////28H MAPPING FUNCTION PARAMETERS,
     1        13H FOR ZONE NO.,I2)
   30 FORMAT (28H **NO MAPPING IN THIS ZONE**)
   40 FORMAT (21H MAPPING FUNCTION NO.,I2,8H IN ZONE/
     1        13H MAPPING TYPE,I2/12H PARAMETERS:,5E12.5/)
      END


      SUBROUTINE PRINTR (IZN)
C
C         ROUTINE TO PRINT COEFFICIENTS
C         OF EITHER A PHI OR PSI FUNCTION
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      COMMON   /SOLN/  A(15401),IDIAG(176),PHI(176)
      DATA ZERO /0./
      IZR=IZRO(IZN)
      IOR=IODRL(IZN)
      IOI=IODIM(IZN)
      IER=IEVRL(IZN)
      IEI=IEVIM(IZN)
      PRINT 40
      NPA=NNEG(IZN)
      NP=-NPA
      NPWRS=NPA+NPOS(IZN)+1
      IEO=2
      IF (NPA/2.NE.(NPA+1)/2)  IEO=1
      IND=IBEGI(IZN)
      DO 30  I=1,NPWRS
      PRTRL=ZERO
      PRTIM=ZERO
```

```
      IF (NP.EQ.0.AND.IZR.EQ.0)  GO TO 20
      IF ((IOR.EQ.0.AND.IEO.EQ.1).OR.(IER.EQ.0.AND.IEO.EQ.2))  GO TO 10
      PRTRL=PHI(IND)
      IND=IND+1
   10 IF ((IOI.EQ.0.AND.IEO.EQ.1).OR.(IEI.EQ.0.AND.IEO.EQ.2))  GO TO 20
      PRTIM=PHI(IND)
      IND=IND+1
   20 PRINT 50, NP,PRTRL,PRTIM
      NP=NP+1
      IEO=3-IEO
   30 CONTINUE
      RETURN
   40 FORMAT (4H POW,4X,9HREAL COEF,4X,9HIMAG COEF )
   50 FORMAT (1H ,I3,2E13.5)
      END


      SUBROUTINE PRINTS (NZ)
C
C        ROUTINE TO PRINT INFORMATION ABOUT
C        EITHER A PHI OR PSI FUNCTION IN A ZONE
C
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      IF (NZ.GT.5)  GO TO 10
      PRINT 30, NZ
      GO TO 20
   10 NNZ=NZ-5
      PRINT 40, NNZ
   20 NNN=-NNEG(NZ)
      IF (NNEG(NZ).GT.0)  PRINT 50, NNN
      IF (NPOS(NZ).GT.0)  PRINT 60, NPOS(NZ)
      IF (IZRO(NZ).GT.0)  PRINT 70
      PRINT 80, IODRL(NZ),IODIM(NZ),IEVRL(NZ),IEVIM(NZ)
      RETURN
   30 FORMAT (//22H PHI FUNCTION FOR ZONE,I2)
   40 FORMAT (//22H PSI FUNCTION FOR ZONE,I2)
   50 FORMAT (26H HIGHEST NEGATIVE POWER IS,I4)
   60 FORMAT (26H HIGHEST POSITIVE POWER IS,I4)
   70 FORMAT (51H ZERO POWER INCLUDED(IF CONSISTENT WITH SYMMETRIES))
   80 FORMAT (34H COEFFICIENTS INCLUDED(1=YES,0=NO)/
     1         11H  ODD POWER,5X,10HEVEN POWER/11H REAL  IMAG,5X,
     2         10HREAL  IMAG/1X,I4,I6,5X,I4,I6)
      END


      SUBROUTINE PRNTMT (IZ)
C
C        ROUTINE TO PRINT MATERIAL PROPERTIES
C
      COMMON   /MAT/  G(5),ETAM(5)
      PRINT 10, IZ
      PRINT 20, G(IZ),ETAM(IZ)
      RETURN
   10 FORMAT (////33H MATERIAL PARAMETERS FOR ZONE NO.,I2)
   20 FORMAT (6H   G =,E12.5/6H ETA =,E12.5)
      END


      SUBROUTINE PRNTSR (IZ)
C
C        ROUTINE TO PRINT INFORMATION ON
C        ALL SERIES IN A ZONE
C
```

```
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      PRINT 20, IZ
      NZ=IZ
      DO 10  I=1,2
      IF (ISTYP(NZ).LE.0)  GO TO 10.
      CALL PRINTS (NZ)
   10 NZ=NZ+5
      RETURN
   20 FORMAT (////32H SERIES INFORMATION FOR ZONE NO.,I2)
      END


      SUBROUTINE PRTCOF (IOPT)
C          SUBROUTINE TO PRINT COEFFICIENTS OF
C            PHI AND PSI FUNCTIONS IN VARIOUS ZONES
C               IOPT - INTEGER CONTAINING ZONE NUMBERS
C
C
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      DIMENSION IZONE(5)
      CALL DECODE (IOPT,5,IZONE,NZNZ)
      IF (NZNZ.EQ.0)  RETURN
      PRINT 50
      DO 20  I=1,NZNZ
      NZ=IZONE(NZNZ-I+1)
      IF (NPOS(NZ).LE.0.AND.NNEG(NZ).LE.0)  GO TO 10
      ISER=ISTYP(NZ)
      PRINT 30, NZ
      PRINT 60, ISER
      CALL PRINTR (NZ)
   10 NZ=NZ+5
      IF (NPOS(NZ).LE.0.AND.NNEG(NZ).LE.0)  GO TO 20
      ISER=ISTYP(NZ)
      PRINT 40
      PRINT 60, ISER
      CALL PRINTR (NZ)
   20 CONTINUE
      RETURN
   30 FORMAT (///9H ZONE NO.,I2//13H PHI FUNCTION )
   40 FORMAT (//13H PSI FUNCTION )
   50 FORMAT (//////34H COEFFICIENTS IN PHI,PSI FUNCTIONS  )
   60 FORMAT (12H SERIES TYPE,I2/)
      END


      SUBROUTINE PRTCON (IOPT)
C
C           ROUTINE TO PRINT EXTRA PARAMETERS
C             IOPT - VARIABLE CONTAINING PARAMETER NUMBERS
C
      COMMON  /CONST/ ICON(5,2),ISTRT(5)
      COMMON  /SOLN/  A(15401),IDIAG(176),PHI(176)
      DIMENSION IPRT(5)
      CALL DECODE (IOPT,5,IPRT,NC)
      IF (NC.EQ.0)  GO TO 40
      DO 30  I=1,NC
      IC=IPRT(NC-I+1)
      IF (ICON(IC,1).EQ.0.AND.ICON(IC,2).EQ.0)  GO TO 30
      CONRL=0.
      CONIM=0.
      I1=ISTRT(IC)
      IF (ICON(IC,1).EQ.0)  GO TO 10
```

```
      CONRL=PHI(I1)
      I1=I1+1
   10 IF (ICON(IC,2).EQ.0)  GO TO 20
      CONIM=PHI(I1)
   20 PRINT 50, IC,CONRL,CONIM
   30 CONTINUE
   40 RETURN
   50 FORMAT (///20H EXTRA PARAMETER NO.,I2/1X,2E13.5)
      END


      SUBROUTINE PRTZON (IOPT)
C
C        ROUTINE TO PRINT ALL INFORMATION IN ZONE
C            IOPT - VARIABLE CONTAINING ZONE NUMBERS
C
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      DIMENSION IZONE(5)
      CALL DECODE (IOPT,5,IZONE,NZNZ)
      IF (NZNZ.EQ.0)  RETURN
      PRINT 20
      DO 10  I=1,NZNZ
      IZ=IZONE(NZNZ-I+1)
      IF (IZ.GT.5.OR.ISTYP(IZ).LE.0.AND.ISTYP(IZ+5).LE.0)  RETURN
      CALL PRNTSR (IZ)
      CALL PRNTMT (IZ)
   10 CALL PRNTMP (IZ)
      RETURN
   20 FORMAT (1H1)
      END


      SUBROUTINE PTCOL (XP,YP,IZ,IOPT,IREF,WT,IF1,IF2,IT1,IT2)
C
C
C          SUBROUTINE TO COLLOCATE AT ONE POINT EITHER FORCES
C           OR DISPLACEMENTS
C
C                IOPT      - 1--COLLOCATE FORCES
C                            2--COLLOCATE DISPLACEMENTS
C                IREF      - 1--NO REFLECTIONS
C                            2--X-AXIS REFLECTION
C                            3--UNIT CIRCLE REFLECTION
C                WT        - WEIGHTING OF EQUATIONS
C                IF1,IF2   - WHICH EQUATIONS TO SAVE
C                ITO1,ITO2- WHERE TO PLACE EQUATIONS TO BE SAVED
C                             IN MASTER EQUATIONS(EITHER 1ST OR 2ND EQUAT
C
C
      COMMON    /MAT/  G(5),ETAM(5)
      C1=1.
      C2=1.
      IF (IOPT.EQ.1)  GO TO 10
      C1=ETAM(IZ)
      C2=-1.
   10 CALL ZERO (1,2,0,0,0,0)
      CALL B4PHIP (XP,YP,IZ,IREF,C2,A,B)
      CALL PHIPB (XP,YP,A,B,IZ)
      A=1.
      B=0.
      IF(IREF.EQ.4)CALL MAPPR(XP,YP,A,B,IZ)
      CALL PHI (XP,YP,C1*A,-C1*B,IZ)
      GO TO (40,20,30,40), IREF
   20 CALL PHIB (XP,YP,-C2,0.,IZ)
```

```
      GO TO 40
   30 CALL PHIB1 (XP,YP,-C2,0.,IZ)
   40 CALL PLACE (IZ,IF1,IF2,IT1,IT2,WT)
      IF (IREF.EQ.2.OR.IREF.EQ.3)  GO TO 50
      CALL ZERO (1,2,0,0,0,0,0)
      CALL PHIBR (XP,YP,C2,0.,IZ+5)
      CALL PLACE (IZ+5,IF1,IF2,IT1,IT2,WT)
   50 RETURN
      END


      SUBROUTINE PWSERD(XP,YP,A,B,IZ)
C
C          COMPUTER TERMS IN DERIVATIVE
C          OF POWER SERIES
C
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      COMMON /POWERS/ R(100),TH(100),ISHFT
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      NTOT1=NNEG(IZ)+NPOS(IZ)+1
      IPOW=-NNEG(IZ)
      XX=XP-XNOT(IZ)
      YY=YP-YNOT(IZ)
      XIPOW=IPOW
      RHOAB=SQRT(A*A+B*B)
      IF(RHOAB.NE.0.)PHIAB=ATAN2(B,A)
      R1=SQRT(XX*XX+YY*YY)
      TH1=ATAN2(YY,XX)
      DO 10 I=1,NTOT1
      I1=ISHFT+I
      XIPOW=IPOW
      R(I1)= RHOAB*XIPOW*(R1**(IPOW-1))
      TH(I1)=BRANCH(PHIAB+(XIPOW-1.)*TH1,-PI)
      IF(IPOW.GE.0)GO TO 10
      R(I1)=-R(I1)
      TH(I1)=BRANCH(TH(I1)+PI,-PI)
   10 IPOW=IPOW+1
      RETURN
      END



      SUBROUTINE QUADF (A,B,PHI,X0,Y0,XM,YM,IMAP,XQ,YQ)
C
C
C            SUBROUTINE USED BY INVERSE MAPPINGS TO COMPUTE SQARE ROOT
C            OF FUNCTION OF THE FORM (Z*Z -C*C)   --  C IS REAL
C
C
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      EQUIVALENCE (PIH,HLFPI), (PIT,TWOPI)
      DENOM=A+B
      X1=XM-X0
      Y1=YM-Y0
      IF (DENOM.EQ.0)  GO TO 50
      DENOM=1./DENOM
      FACT=1.
      IF (IMAP.EQ.-2)  FACT=-1.
      ICT=0
      CS=COS(PHI)
      SN=SIN(PHI)
      XZ=X1*CS+Y1*SN
      YZ=Y1*CS-X1*SN
      XC=0.
```

```
      YC=0.
      AA=A*A
      BB=B*B
      IF (AA.GT.BB)  XC=SQRT(AA-BB)
      IF (BB.GT.AA)  YC=SQRT(BB-AA)
      X1=XZ+XC
      Y1=YZ+YC
      R1=SQRT(X1*X1+Y1*Y1)
      TH1=0.
      IF (X1.EQ.0..AND.Y1.EQ.0.)  GO TO 10
      TH1=ATAN2(Y1,X1)
      IF (BB.GT.AA)  TH1=BRANCH(TH1,-3.*PI)
   10 X1=XZ-XC
      Y1=YZ-YC
      R2=SQRT(X1*X1+Y1*Y1)
      TH2=0.
      IF (X1.EQ.0..AND.Y1.EQ.0.)  GO TO 20
      TH2=ATAN2(Y1,X1)
      IF (BB.GT.AA)  TH2=BRANCH(TH2,-PIH)
      IF (AA.GT.BB)  TH2=BRANCH(TH2,0.)
   20 X2=SQRT(R1*R2)
      Y2=.5*(TH1+TH2)
   30 XQ=DENOM*(XZ+FACT*X2*COS(Y2))
      YQ=DENOM*(YZ+FACT*X2*SIN(Y2))
      XX=XQ*XQ+YQ*YQ
      IF (XX.GT..9999.OR.IABS(IMAP).EQ.2)  RETURN
      IF (ICT.NE.1)  GO TO 40
      WRITE (6,60) XM,YM
      STOP
   40 FACT=-FACT
      ICT=1
      GO TO 30
   50 XQ=ATAN2(Y1,X1)
      IF (XQ.GT.PHI+PI)  XQ=XQ-PIT
      IF (XQ.LT.PHI-PI)  XQ=XQ+PIT
      XQ=PHI-XQ
      YQ=ALOG(SQRT(X1*X1+Y1*Y1)/A)
      RETURN
   60 FORMAT (15H ERROR IN QUADF,10X,3HXM=,E12.4,5X,3HYM=,E12.4)
      END


      SUBROUTINE PWSRD2(XP,YP,A,B,IZ)
C
C          ROUTINE TO COMPUTE TERMS IN
C          SECOND DERIVATIVE OF POWER SERIES
C
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      COMMON /POWERS/ R(100),TH(100),ISHFT
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      NTOT1=NNEG(IZ)+NPOS(IZ)+1
      IPOW=-NNEG(IZ)
      XX=XP-XNOT(IZ)
      YY=YP-YNOT(IZ)
      R1=SQRT(XX*XX+YY*YY)
      TH1=ATAN2(YY,XX)
      RHOAB=SQRT(A*A+B*B)
      PHIAB=ATAN2(B,A)
      DO 10 I=1,NTOT1
      XIPOW=IPOW-2
      RFACT=IPOW*(IPOW-1)
      I1=ISHFT+I
      R(I1)=RHOAB*RFACT*(R1**(IPOW-2))
      TH(I1)=BRANCH(PHIAB+XIPOW*TH1,-PI)
   10 IPOW=IPOW+1
      RETURN
      END
```

```
      SUBROUTINE RADD (RR,TT,NTRMS)
C
C        ADD SERIES OF COMPLEX NUMBERS IN POLAR FORM
C
      COMMON /POWERS/ R(100),TH(100),ISHFT
      DIMENSION RR(1), TT(1)
      DO 30  I=1,NTRMS
      I1=ISHFT+I
      XX=RR(I1)*COS(TT(I1))+R(I1)*COS(TH(I1))
      YY=RR(I1)*SIN(TT(I1))+R(I1)*SIN(TH(I1))
      IF (XX.NE.0..OR.YY.NE.0.)  GO TO 10
      R(I1)=0.
      TH(I1)=0.
      GO TO 30
   10 TH(I1)=ATAN2(YY,XX)
      IF (ABS(COS(TH(I1))).LT..1)  GO TO 20
      R(I1)=XX/COS(TH(I1))
      GO TO 30
   20 R(I1)=YY/SIN(TH(I1))
   30 CONTINUE
      RETURN
      END


      SUBROUTINE RHS (IOPT,NPT,X,Y,X1,Y1,X2,Y2,RHS10,RHS20,SIG0,SIG1,
     1                RHS1,RHS2)
C
C          ROUTINE TO COMPUTE VALUES OF F1 + I F2
C            MEANING OF IOPT
C                1 - CONSTANT VALUE
C                2 - LINEARLY VARYING NORMAL STRESS, STRAIGHT BOUNDARY
C                3 - LINEARLY VARYING SHEAR STRESS,STRAIGHT BOUNDARY
C                4 - CONSTANT NORMAL STRESS, CIRCULAR,ELLIPTICAL BOUNDARY
C                5 - CONSTANT SHEAR STRESS, CIRCULAR,ELLIPTICAL BOUNDARY
C
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      DIMENSION X(1), Y(1), RHS1(1), RHS2(1)
      C1=0.
      C2=0.
      C3=0.
      C4=0.
      GO TO (90,10,20,60,70), IOPT
   10 C1=1.
      C2=1.
      GO TO 30
   20 C3=1.
      C4=1.
   30 THETAC=ATAN2(Y2-Y1,X2-X1)
      IF (ABS(THETAC-HLFPI).GT..00001)  GO TO 40
      CSINV=1.
      GO TO 90
   40 IF (ABS(THETAC+HLFPI).GT..00001)  GO TO 50
      CSINV=-1.
      GO TO 90
   50 CSINV=1./COS(THETAC)
      GO TO 90
   60 C1=1.
      GO TO 80
   70 C3=1.
   80 CSINV=1.
   90 DO 100  I=1,NPT
      XSQDIF=.5*(X(I)*X(I)-X1*X1)*SIG1*CSINV
      YSQDIF=.5*(Y(I)*Y(I)-Y1*Y1)*SIG1*CSINV
      XDIFF=(X(I)-X1)*SIG0
      YDIFF=(Y(I)-Y1)*SIG0
      RHS1(I)=RHS10+C1*XDIFF+C2*XSQDIF-C3*YDIFF-C4*YSQDIF
      RHS2(I)=RHS20+C1*YDIFF+C2*YSQDIF+C3*XDIFF+C4*XSQDIF
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ROTEQU (IOPT,IZNS,A1,A2,A3,A4)
C
C        MULTIPLY EQUATIONS IN GIVEN
C        ZONES BY COMPLEX CONSTANTS
C
      DIMENSION IZ(5)
      CALL DECODE (IZNS,5,IZ,MAX)
      IF (MAX.LE.0)  RETURN
      DO 10  I=1,MAX
   10 CALL ROTZON (IOPT,IZ(I),A1,A2,A3,A4)
      RETURN
      END



      SUBROUTINE ROTZON(IOPT,IZ,A1,A2,A3,A4)
C
C        MULTIPLY SERIES BY COMPLEX NUMBERS
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      NTRMS=IENDI(IZ)-IBEGI(IZ)+1
      DO 30 I=1,NTRMS
      I1=IBEGI(IZ)+I-1
      IF(IOPT.EQ.2)GO TO 10
      EQU(1,I)=EQUTOT(1,I1)*A1+EQUTOT(2,I1)*A2
   10 IF(IOPT.EQ.1)GO TO 20
      EQU(2,I)=EQUTOT(1,I1)*A3+EQUTOT(2,I1)*A4
   20 IF(IOPT.NE.2)EQUTOT(1,I1)=0.
   30 IF(IOPT.NE.1)EQUTOT(2,I1)=0.
      I1=1
      I2=2
      IF(IOPT.EQ.1)I2=1
      IF(IOPT.EQ.2)I1=2
      CALL PLACE(IZ,I1,I2,I1,I2,1.)
      RETURN
      END


      SUBROUTINE SAVE (IOPT)
C
C
C           SUBROUTINE TO SAVE EQUATIONS STORED IN ARRAY EQUTOT
C                IF IOPT=0 DON'T DO ANY SCALING
C
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      IF (IOPT.EQ.0)  GO TO 20
      DO 10  I=1,2
      DO 10  J=1,NUNK
   10 IF (ABS(EQUTOT(I,J)).GT.SCALE(J))  SCALE(J)=ABS(EQUTOT(I,J))
   20 WRITE (IUNIT) ((EQUTOT(I,J),J=1,NN),I=1,2)
      NWRT=NWRT+1
      RETURN
      END



      SUBROUTINE SCLFIX
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      DO 10  I=1,NUNK
   10 SCALE(I)=1./SCALE(I)
      RETURN
      END
```

```
      SUBROUTINE SERINF(IZ,ISERTP,ISTP,INEG,IPOS,IZERO,ICOEF,
     *                  IPL,X0,Y0,E,XNU)
C
C
C          SUBROUTINE TO SET VARIABLES CONTAINING SERIES
C           INFORMATION
C
C               ISERTP       - 1 FOR PHI FUNCTION
C                              2 FOR PSI FUNCTION
C               ISTP         - EXPANSION TYPE
C               INEG         - NUMBER OF NEGATIVE POWERS
C               IPOS         - NUMBER OF POSITIVE POWERS
C               IZERO        - 0 IF ZERO POWER EXCLUDED
C               IOR          - 0 TO EXCLUDE REAL COEF, ODD POWER
C               IOI          - 0 TO EXCLUDE IMAG COEF, ODD POWER
C               IER          - 0 TO EXCLUDE REAL COEF, EVEN POWER
C               IEI          - 0 TO EXCLUDE IMAG COEF, EVEN POWER
C               IPL          - 1 FOR PLANE STRESS
C                              2 FOR PLANE STRAIN
C               (X0,Y0)      - EXPANSION POINT
C               E,XNU        - ELASTIC CONSTANTS
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON   /MAT/  G(5),ETAM(5)
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      DIMENSION ICOF(4)
      EQUIVALENCE (ICOF(4),IOR),(ICOF(3),IOI),(ICOF(2),IER),
     *            (ICOF(1),IEI)
      IZ1=IZ
      IF (ISERTP.EQ.2)  IZ1=IZ+5
      IZONES=MAX0(IZ,IZONES)
      ISTYP(IZ1)=ISTP
      NNEG(IZ1)=INEG
      NPOS(IZ1)=IPOS
      IZRO(IZ1)=IZERO
      CALL DECODE(ICOEF,4,ICOF,NDUM)
      IODRL(IZ1)=IOR
      IODIM(IZ1)=IOI
      IEVRL(IZ1)=IER
      IEVIM(IZ1)=IEI
      XNOT(IZ1)=X0
      YNOT(IZ1)=Y0
      NEVP=INEG/2+IPOS/2
      NODP=NEVP
      IF (INEG/2.NE.(INEG+1)/2)  NODP=NODP+1
      IF (IPOS/2.NE.(IPOS+1)/2)  NODP=NODP+1
      IF (IZERO.NE.0)  NEVP=NEVP+1
      NZUNK=NEVP*(IER+IEI)+NODP*(IOR+IOI)
      IBEGI(IZ1)=NUNKS+1
      NUNKS=NUNKS+NZUNK
      IENDI(IZ1)=NUNKS
      NUNK=NUNKS
      NN=NUNK+1
      G(IZ)=E/(2.*(1.+XNU))
      GO TO (10,20), IPL
   10 ETAM(IZ)=(3.-XNU)/(1.+XNU)
      GO TO 30
   20 ETAM(IZ)=3.-4.*XNU
   30 RETURN
      END
```

```
      SUBROUTINE SERFIL (IZ,F1,F2,A,B)
C
C
C          SUBROUTINE TO FILL EQUATIONS FOR UNKNOWN COEFFICIENTS
C
C
      COMMON    /EQN/   EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                  SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON /POWERS/ R(100),TH(100),ISHFT
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                  NPOS(10),IZRO(10),IODRL(10),
     *                  IODIM(10),IEVRL(10),IEVIM(10)
      IZR=IZRO(IZ)
      NT1=NNEG(IZ)+NPOS(IZ)+1
      IOR=IODRL(IZ)
      IOI=IODIM(IZ)
      IER=IEVRL(IZ)
      IEI=IEVIM(IZ)
      IPOW=-NNEG(IZ)
      IEO=1
      IP1=ABS(IPOW)
      IF (IP1/2.EQ.(IP1+1)/2)  IEO=2
      IND=1
      DO 30  I=1,NT1
      IF (IZR.EQ.0.AND.IPOW.EQ.0)  GO TO 20
      CS=COS(TH(I))
      SN=SIN(TH(I))
      IF ((IOR.EQ.0.AND.IEO.EQ.1).OR.(IER.EQ.0.AND.IEO.EQ.2))  GO TO 10
      EQU(1,IND)=EQU(1,IND)+R(I)*(A*CS-B*F1*SN)
      EQU(2,IND)=EQU(2,IND)+R(I)*(B*CS+A*F1*SN)
      IND=IND+1
   10 IF ((IOI.EQ.0.AND.IEO.EQ.1).OR.(IEI.EQ.0.AND.IEO.EQ.2))  GO TO 20
      EQU(1,IND)=EQU(1,IND)-F2*R(I)*(B*CS+A*F1*SN)
      EQU(2,IND)=EQU(2,IND)+F2*R(I)*(A*CS-B*F1*SN)
      IND=IND+1
   20 IPOW=IPOW+1
      IEO=3-IEO
   30 CONTINUE
      RETURN
      END


      SUBROUTINE SLVINT
C
C
C         SUBROUTINE TO INITIALIZE VARIABLES FOR SOLUTION ROUTINE
C
C
      COMMON    /EQN/   EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                  SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON    /SOLN/  A(15401),IDIAG(176),PHI(176)
      IDIAG(1)=1
      N=NUNK
      DO 10  I=1,NUNK
      IDIAG(I+1)=IDIAG(I)+N
   10 N=N-1
      N=IDIAG(NUNK)
      DO 20  I=1,N
   20 A(I)=0.
      DO 30  I=1,NUNK
   30 PHI(I)=0.
      RETURN
      END
```

```
      SUBROUTINE STCOL(XP,YP,N,IZ,IREF,ALPHA,RHS1,RHS2,IERR)
C
C
C           SUBROUTINE TO COMPUTE STRESS EQUATIONS
C                MEANING OF IREF
C
C                  1 - NO REFLECTIONS
C                  2 - X-AXIS REFLECTION
C                  3 - UNIT CIRCLE REFLECTION
C             ALPHA - ANGLE OF NORMAL TO SURFACE ON WHICH TO FIND STRESS
C
C
      DIMENSION XP(1),YP(1),RHS1(1),RHS2(1),ALPHA(1)
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON   /PIFAC/ PI,HLFPI,TWOPI
      DO 20 I=1,N
      CALL ZERO(0,0,1,2,1,0)
      ALFA=PI*ALPHA(I)/180.
      CALL STRPT(XP(I),YP(I),IZ,ALFA,IREF,1.,1,2,1,2)
      EQUTOT(1,NN)=RHS1(I)
      EQUTOT(2,NN)=RHS2(I)
      IF(IERR.EQ.1)GO TO 10
      CALL SAVE(1)
      GO TO 20
   10 CALL EVAL(1,2,2,XER,YER)
      PRINT 30, XER,YER,XP(I),YP(I)
   20 CONTINUE
      RETURN
   30 FORMAT(1H ,2F10.4,2E10.4)
       END

      SUBROUTINE SOLVE
C
C
C           SUBROUTINE TO SOLVE SIMULTANEOUS EQUATIONS
C
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON   /SOLN/  A(15401),IDIAG(176),PHI(176)
      EQUIVALENCE (N,NUNK)
      DIMENSION T(176)
      DO 70  I=1,N
      L=IDIAG(I)
      TEMP=A(L)
      LL=IDIAG(I+1)-1
      II=I-L
      DO 10  J=L,LL
      T(II+J)=A(J)
   10 A(J)=A(J)/TEMP
      PHI(I)=PHI(I)/TEMP
      DO 60  J=1,N
      MM=IDIAG(J+1)-1
      NOFF=LL-MM
      IF (J-I)  20,60,30
   20 M=IDIAG(J)+I-J
      TEMP=A(M)
      GO TO 40
   30 M=IDIAG(J)
      TEMP=T(J)
   40 DO 50  K=M,MM
   50 A(K)=A(K)-TEMP*A(K+NOFF)
      PHI(J)=PHI(J)-TEMP*PHI(I)
   60 CONTINUE
   70 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STITCH (NZ1,NZ2,NREF1,NREF2,ISOPT,INV,
     1                   INVZN,NPT,XX,YY,F1,F2,U,V,IERR)
C
C
C          SUBROUTINE TO STITCH TWO ZONES
C               ISOPT          - 1--STITCH FORCES ONLY
C                                2--STITCH DISPLACEMENTS ONLY
C                                3--STITCH FORCES,DISPLACEMENTS
C               INV            - 1--POINTS COMING FROM PARAMETER PLANE
C                                   OF ZONE INVZN
C               IERR           - 0 IF NOT CALCULATING ERRORS
C
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      COMMON   /MAT/  G(5),ETAM(5)
      DIMENSION XX(1), YY(1), F1(1), F2(1), U(1), V(1)
      IREF1=NREF1
      IREF2=NREF2
      IZ1=NZ1
      IZ2=NZ2
      IF (INV.EQ.2.OR.INVZN.EQ.NZ1.AND.INV.EQ.1)  GO TO 10
      IREF1=NREF2
      IREF2=NREF1
      IZ1=NZ2
      IZ2=NZ1
   10 IFU1=1
      IFU2=2
      IF (ISOPT.EQ.1)  IFU2=1
      IF (ISOPT.EQ.2)  IFU1=2
      DO 90  I=1,NPT
      GO TO (20,30), INV
   20 XP1=XX(I)
      YP1=YY(I)
      GO TO 40
   30 X=XX(I)
      Y=YY(I)
   40 CALL MAPPER (XP1,YP1,X,Y,IZ1,INV)
      CALL MAPPER (XP2,YP2,X,Y,IZ2,2)
      DO 90  J=IFU1,IFU2
      FACT=-1.
      IF(J.EQ.2)FACT=-G(IZ1)/G(IZ2)
      CALL ZERO (0,0,1,2,1,0)
      CALL PTCOL (XP1,YP1,IZ1,J,IREF1,1.,1,2,1,2)
      IF(IREF1.EQ.4)CALL MLTDSP(XP1,YP1,IZ1)
      CALL PTCOL (XP2,YP2,IZ2,J,IREF2,FACT,1,2,1,2)
      IF(IREF2.EQ.4)CALL MLTDSP(XP2,YP2,IZ2)
      GO TO (50,60), J
   50 EQUTOT(1,NN)=F1(I)
      EQUTOT(2,NN)=F2(I)
      GO TO 70
   60 EQUTOT(1,NN)=U(I)
      EQUTOT(2,NN)=V(I)
   70 IF (IERR.EQ.0)  GO TO 80
      CALL EVAL (1,2,2,XER,YER)
      PRINT 100, XER,YER,XP1,YP1,XP2,YP2
      GO TO 90
   80 CALL SAVE (1)
   90 CONTINUE
      RETURN
  100 FORMAT (1H ,2F10.4,4E10.4)
      END

      SUBROUTINE STRPT(XP,YP,IZ,ALFA,IREF,WT,IF1,IF2,IT1,IT2)
C
C
C          SUBROUTINE TO COLLOCATE STRESSES AT A POINT
C              XP + I YP  - POINT AT WHICH TO COLLOCATE
```

```
C              IZ           - ZONE NUMBER
C              ALFA         - ANGLE OF NORMAL TO SURFACE AT WHICH STRESS I
C              IREF         - REFLECTION OPTION
C                                 1 -- NO REFLECTIONS
C                                 2 -- X-AXIS REFLECTION
C                                 3 -- UNIT CIRCLE REFLECTION
C              WT           - WEIGHTING OF EQUATIONS
C
C
      ALPHA=-ALFA
      CSA=COS(2.*ALPHA)
      SNA=SIN(2.*ALPHA)
      CALL ZERO(1,2,0,0,0,0)
      IF(IF1.GT.1)GO TO 10
      CALL MAPPR(XP,YP,A,B,IZ)
      DENOM=1./(A*A+B*B)
      AA=2.*A*DENOM
      BB=-2.*B*DENOM
      CALL PHIPB(XP,YP,AA,-BB,IZ)
      CALL ZERO(2,2,0,0,0,0)
   10 CALL B4PHDP(XP,YP,IZ,CSA,SNA,IREF,A,B)
      CALL PHIPPB(XP,YP,A,-B,IZ)
      CALL B4PHP(XP,YP,CSA,SNA,IZ,IREF,A,B)
      CALL PHIPB(XP,YP,A,-B,IZ)
      CALL MAPPR(XP,YP,XPR,YPR,IZ)
      CSA=COS(2.*ALPHA)
      SNA=SIN(2.*ALPHA)
      DENOM=1./(XPR*XPR+YPR*YPR)
      A=DENOM*(CSA*XPR+SNA*YPR)
      B=DENOM*(SNA*XPR-CSA*YPR)
      GO TO (40,20,30),IREF
   20 CALL PHIBP1(XP,YP,A,-B,IZ)
      GO TO 40
   30 XT=XP*XP-YP*YP
      YT=2.*XP*YP
      DENOM=1./(XT*XT+YT*YT)
      AA=-DENOM*(A*XT+B*YT)
      BB= DENOM*(A*YT-B*XT)
      CALL PHIP1B(XP,YP,AA,-BB,IZ)
   40 CALL PLACE(IZ,IF1,IF2,IT1,IT2,WT)
      IF(IREF.NE.1)GO TO 50
      A=-A
      B=-B
      CALL ZERO(1,2,0,0,0,0)
      CALL PHIPB(XP,YP,A,-B,IZ+5)
      CALL PLACE(IZ+5,IF1,IF2,IT1,IT2,WT)
   50 RETURN
      END

      SUBROUTINE STRRES (IZ,NPT,IREF,ALFA,X,Y,XP,YP)
C
C
C             SUBROUTINE TO COMPUTE AND PRINT STRESSES FROM SOLUTION
C
C                IREF          - 1 FOR NO REFLECTIONS
C                                2 FOR X-AXIS REFLECTION
C                                3 FOR UNIT CIRCLE REFLECTION
C                ALFA          - ANGLES OF NORMAL TO SURFACES ON WHICH ST
C                                ARE CALCULATED
C
C
      DIMENSION X(1), Y(1), XP(1), YP(1), ALFA(1)
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      FACT=180./PI
      PRINT 20
      DO 10  I=1,NPT
      CALL ZERO (0,0,1,2,3,0)
      CALL STRPT (XP(I),YP(I),IZ,ALFA(I),IREF,1.,1,2,1,2)
```

```
      CALL EVAL (1,2,1,SIG,TAU)
      CALL ZERO(0,0,1,2,3,0)
      CALL STRPT(XP(I),YP(I),IZ,ALFA(I)+HLFPI,IREF,1.,1,1,1,1)
      CALL EVAL(1,1,1,SIG90,SDUM)
      ALF=ALFA(I)*FACT
   10 PRINT 30, SIG,SIG90,TAU,ALF,X(I),Y(I),XP(I),YP(I)
      RETURN
   20 FORMAT (////23H0EVALUATION OF STRESSES / 7X,7HSTRESS1,
     *        6X,7HSTRESS2,1X,12HSHEAR STRESS,5X,5HANGLE,
     *        9X,1HX,12X,1HY,8X,8HX(PARAM),5X,8HY(PARAM) )
   30 FORMAT (1X,3E13.5,F10.3,4E13.5)
      END


      FUNCTION TANGO (X1,X2)
      TANGO=BRANCH(ATAN2(X1,X2),0.)
      RETURN
      END


      SUBROUTINE UNIT(I)
C
C     SUBROUTINE TO SET UP NEW TAPE FOR WRITING EQUATIONS
C
      COMMON    /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                 SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      IUNIT=I
      NWRT=0
      RETURN
      END


      SUBROUTINE WRTDSP(NFB,NFD,IZ,IREF)
C
C        ROUTINE TO WRITE DISPLACEMENTS AT INTERFACE
C        OF MMC AND FINITE ELEMENT REPRESENTATIONS FOR
C        BACKSUBSTITUTION INTO FINITE ELEMENT EQUATIONS
C
      COMMON    /MAT/   G(5),ETAM(5)
      COMMON    /MMCFE/ X(150),Y(150),FDRHS(300),EQNS(120,176),
     *                  KUV(120,2)
      DIMENSION FD(1)
      EQUIVALENCE (FD(1),FDRHS(1))
      REWIND NFB
      REWIND NFD
      READ (NFB) N1,N2,IDOFST
      NPT=N2/2
      READ (NFB) (X(I),Y(I),I=1,NPT)
      A=1.
      B=0.
      DO 100 I=1,NPT
      CALL ZERO(0,0,1,2,1,0)
      CALL MAPPER(XP,YP,X(I),Y(I),IZ,2)
      CALL PTCOL(XP,YP,IZ,2,IREF,1.,1,2,1,2)
      IF(IREF.EQ.4)CALL MAPPR(XP,YP,A,B,IZ)
      FACT=1./((A*A+B*B)*2.*G(IZ))
      CALL EVAL(1,2,1,V1,V2)
      FD(2*I-1)=(V1*A-V2*B)*FACT
  100 FD(I*2  )=(V1*B+V2*A)*FACT
      WRITE (NFD) (FD(I),I=1,N2)
      END FILE NFD
      RETURN
      END
```

```
      SUBROUTINE ZETN(XP,YP,IZ)
C
C        ROUTINE TO CALCULATE TERMS
C        IN EXPANSION OF PHI, PSI
C
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      ITYP=ISTYP(IZ)
      GO TO (10,20),ITYP
   10 A=1.
      B=0.
      GO TO 40
   20 YY=YP+1.
      DEN=1./(XP*XP+YY*YY)
      A= XP*DEN
      B=-YY*DEN
   40 CALL POWSER(XP,YP,A,B,IZ)
      RETURN
      END


      SUBROUTINE ZERO (IE1,IE2,IS1,IS2,IOPT,IINPT)
C
C
C           SUBROUTINE TO ZERO OUT ZONE AND MASTER EQUATIONS
C
C              IE1,IE2     -BEGINNING AND ENDING EQUATION TO ZERO OUT
C                            (ZONE EQUATIONS)
C              IS1,IS2     -BEGINNING AND ENDING EQUATION TO ZERO OUT
C                            (MASTER EQUATIONS)
C              IOPT        -OPTION FOR ZEROING OUT MASTER EQUATIONS
C                            1--ZERO UP TO CONSTANTS
C                            2--ZERO TO TOTAL UNKNOWNS
C                            3--ZERO TO TOTAL UNKNOWNS +1 (RHS)
C                            4--ZERO TO UNKNOWN IINPT
C
C
      COMMON   /EQN/  EQU(3,100),IBEGI(10),IENDI(10),EQUTOT(2,175),
     *                SCALE(175),NN,NUNKS,NUNK,NWRT,IUNIT
      IF (IE2-IE1.LT.0.OR.(IE2.EQ.0.AND.IE1.EQ.0))  GO TO 20
      DO 10  I=IE1,IE2
      DO 10  J=1,100
   10 EQU(I,J)=0.
   20 IF (IS2-IS1.LT.0.OR.(IS2.EQ.0.AND.IS1.EQ.0))  RETURN
      NT=NUNKS
      IF (IOPT.EQ.2)  NT=NUNK
      IF (IOPT.EQ.3)  NT=NN
      IF (IOPT.EQ.4)  NT=IINPT
      DO 30  I=IS1,IS2
      DO 30  J=1,NT
   30 EQUTOT(I,J)=0.
      RETURN
      END


      SUBROUTINE ZETN1(XP,YP,IZ)
C
C        ROUTINE TO COMPUTE TERMS IN FIRST
C        DERIVATIVE OF EXPANSION FOR PHI, PSI
C
      DIMENSION RT(100),TT(100)
      COMMON /POWERS/ R(100),TH(100),ISHFT
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      NTOT1=NNEG(IZ)+NPOS(IZ)+1
      ITYP=ISTYP(IZ)
```

```
      GO TO (10,20),ITYP
   10 A=1.
      B=0.
      GO TO 40
   20 YP1=YP+1.
      DAB=1./(XP*XP+YP1*YP1)
      DCD=DAB*DAB
      A= XP*DAB
      B=-YP1*DAB
      C=DCD*(YP1*YP1-XP*XP)
      D=DCD*XP*YP1*2.
   40 CALL PWSERD(XP,YP,A,B,IZ)
      IF(ITYP.EQ.1)GO TO 60
      DO 50 I=1,NTOT1
      RT(I)=R(I)
   50 TT(I)=TH(I)
      CALL POWSER(XP,YP,C,D,IZ)
      CALL RADD(RT,TT,NTOT1)
   60 RETURN
      END


      SUBROUTINE ZETN2(XP,YP,IZ)
C
C        ROUTINE TO COMPUTE TERMS IN SECOND
C        DERIVATIVE OF EXPANSION FOR PHI, PSI
C
      DIMENSION RT(100),TT(100)
      COMMON  /PIFAC/ PI,HLFPI,TWOPI
      COMMON /POWERS/ R(100),TH(100),ISHFT
      COMMON /SERIES/ ISTYP(10),IZONES,XNOT(10),YNOT(10),NNEG(10),
     *                NPOS(10),IZRO(10),IODRL(10),
     *                IODIM(10),IEVRL(10),IEVIM(10)
      NTOT1=NNEG(IZ)+NPOS(IZ)+1
      ITYP=ISTYP(IZ)
      GO TO (10,20),ITYP
   10 A=1.
      B=0.
      GO TO 30
   20 YP1=YP+1.
      DEN=1./(XP*XP+YP1*YP1)
      A= XP*DEN
      B=-YP1*DEN
   30 CALL PWSRD2(XP,YP,A,B,IZ)
      IF(ITYP.EQ.1)GO TO 60
      DO 40 I=1,NTOT1
      RT(I)=R(I)
   40 TT(I)=TH(I)
      CALL ZETN1(XP,YP,IZ)
      DEN=2.*SQRT(DEN)
      TTH=-ATAN2(YP1,XP)+PI
      DO 50 I=1,NTOT1
      R(I)=R(I)*DEN
   50 TH(I)=BRANCH(TH(I)+TTH,-PI)
      CALL RADD(RT,TT,NTOT1)
   60 RETURN
      END
```

DISTRIBUTION LIST

No. of Copies	To
1	Office of the Under Secretary of Defense for Research and Engineering, The Pentagon, Washington, D.C. 20301
12	Commander, Defense Documentation Center, Cameron Station, Building 5, 5010 Duke Street, Alexandria, Virginia 22314
1	Metals and Ceramics Information Center, Battelle Columbus Laboratories, 505 King Avenue, Columbus, Ohio 43201
	Deputy Chief of Staff, Research, Development, and Acquisition, Headquarters Department of the Army, Washington, D. C. 20310
1	ATTN: DAMA-ARZ
	Commander, Army Research Office, P. O. Box 12211, Research Triangle Park, North Carolina 27709
1	ATTN: Information Processing Office
1	Dr. F. W. Schmiedeshoff
	Commander, U. S. Army Materiel Development and Readiness Command, 5001 Eisenhower Avenue, Alexandria, Virginia 22333
1	ATTN: DRCLDC, Mr. R. Zentner
	Commander, U. S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, Maryland 21005
1	ATTN: DRXSY-MP, H. Cohen
	Commander, U. S. Army Communications Research and dUvelopment Command, Fort Monmouth, New Jersey 07703
1	ATTN: DELSD-L
1	DELSD-E
1	DRCDO-GG-E
1	DRCDO-GG-EA
1	DRCDO-GG-ES
1	DRCDO-GG-EG
1	DRCDO-GG-EI
	Commander, U. S. Army Missile Research and Development Command, Redstone Arsenal, Alabama 35809
1	ATTN: DRDMI-RKK, Mr. C. Martens, Bldg. 7120
	Commander, U. S. Army Natick Research and Development Command, Natick, Massachusetts 01760
1	ATTN: Technical Library
1	Dr. E. W. Ross
1	DRDNA-UE, Dr. L. A. McClaine
	Commander, U. S. Army Satellite Communications Agency, Fort Monmouth, New Jersey 07703
1	ATTN: Technical Document Center
	Commander, U. S. Army Tank-Automotive Research and Development Command, Warren, Michigan 48090
1	ATTN: DRDTA-RKA
1	DRDTA-UL, Technical Library
	Commander, U. S. Army Armament Research and Development Command, Dover, New Jersey 07801
2	ATTN: Technical Library
1	DRDAR-SCM, J. D. Corrie
1	Dr. J. Fraiser
1	Mr. H. E. Pebly, Jr., PLASTEC, Director
	Commander, White Sands Missile Range, New Mexico 88002
1	ATTN: STEWS-WS-VT
	Commander, Aberdeen Proving Ground, Maryland 21005
1	ATTN: STEAP-TL, Bldg. 305
	Commander, U. S. Army Armament Research and Development Command, Aberdeen Proving Ground, Maryland 21010
1	ATTN: DRDAR-QAC-E
	Commander, U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland 21005
1	ATTN: Dr. R. Vitali
1	Dr. G. L. Filbey
1	Dr. W. Gillich
1	Mr. A. Elder
1	DRDAR-TSB-S (STINFO)
	Commander, Harry Diamond Laboratories, 2800 Powder Mill Road, Adelphi, Maryland 20783
1	ATTN: Technical Information Office
	Commander, Picatinny Arsenal, Dover, New Jersey 07801
1	ATTN: Mr. A. M. Anzalone, Bldg. 3401
1	Mr. J. Pearson
1	G. Randers-Pehrson
1	Mr. A. Garcia
1	SARPA-RT-S
	Commander, Redstone Scientific Information Center, U. S. Army Missile Research and Development Command, Redstone Arsenal, Alabama 35809
1	ATTN: DRDMI-TB
	Commander, Watervliet Arsenal, Watervliet, New York 12189
1	ATTN: Dr. T. Davidson
1	Mr. D. P. Kendall
1	Mr. J. F. Throop
	Commander, U. S. Army Foreign Science and Technology Center, 220 7th Street, N. E., Charlottesville, Virginia 22901
1	ATTN: Mr. Marley, Military Tech
	Chief, Benet Weapons Laboratory, LCWSL, USA ARRADCOM, Watervliet Arsenal, Watervliet, New York 12189
1	ATTN: DRDAR-LCB-TL
	Director, Eustis Directorate, U. S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia 23604
1	ATTN: Mr. J. Robinson, DAVDL-E-MOS (AVRADCOM)

No. of Copies	To
	U. S. Army Aviation Training Library, Fort Rucker, Alabama 36360
1	ATTN: Building 5906-5907
	Commander, U. S. Army Agency for Aviation Safety, Fort Rucker, Alabama 36362
1	ATTN: Librarian, Bldg. 4905
	Commander, USACDC Air Defense Agency, Fort Bliss, Texas 79916
1	ATTN: Technical Library
	Commander, U. S. Army Engineer School, Fort Belvoir, Virginia 22060
1	ATTN: Library
	Commander, U. S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi 39180
1	ATTN: Research Center Library
	Commander, U. S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia 22060
1	ATTN: DRDME-MW, Dr. J. W. Bond
	Commander, Naval Air Engineering Center, Lakehurst, New Jersey 08733
1	ATTN: Technical Library, Code 1115
	Director, Structural Mechanics Research, Office of Naval Research, 800 North Quincy Street, Arlington, Virginia 22203
1	ATTN: Dr. N. Perrone
	Naval Air Development Center, Aero Materials Department, Warminster, Pennsylvania 18974
1	ATTN: J. Viglione
	David Taylor Naval Ship Research and Development Laboratory, Annapolis, Maryland 21402
1	ATTN: Dr. H. P. Chu
	Naval Research Laboratory, Washington, D.C. 20375
1	ATTN: C. D. Beachem, Head, Adv. Mat'ls Tech Br. (Code 6310)
1	Dr. J. M. Krafft - Code 8430
1	E. A. Lange
1	Dr. P. P. Puzak
1	R. J. Sanford - Code 8436
1	A. M. Sullivan
1	R. W. Rice
1	S. W. Freiman
1	Dr. Jim C. I. Chang
	Chief of Naval Research, Arlington, Virginia 22217
1	ATTN: Code 471
	Naval Weapons Laboratory, Washington, D.C. 20390
1	ATTN: H. W. Romine, Mail Stop 103
1	Ship Research Committee, Maritime Transportation Research Board, National Research Council, 2101 Constitution Avenue, N. W., Washington, D.C. 20418
	Air Force Materials Laboratory, Wright-Patterson Air Force Base, Ohio 45433
2	ATTN: AFML (MXE), E. Morrissey
1	AFML (LC)
1	AFML (LLP), D. M. Forney, Jr.
1	AFML (LNC), T. J. Reinhart
1	AFML (MBC), Mr. Stanley Schulman
1	AFFDL (FB), Dr. J. C. Halpin
	Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio 45433
1	ATTN: AFFDL (FBS), C. Wallace
1	AFFDL (FBE), G. D. Sendeckyj
	National Aeronautics and Space Administration, Washington, D.C. 20546
1	ATTN: Mr. B. G. Achhammer
1	Mr. G. C. Deutsch - Code RW
	National Aeronautics and Space Administration, Marshall Space Flight Center, Huntsville, Alabama 35812
1	ATTN: R. J. Schwinghamer, EH01, Dir., M&P Lab
1	Mr. W. A. Wilson, EH41, Bldg. 4612
	National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia 23665
1	ATTN: Mr. H. F. Hardrath, Mail Stop 188M
1	Mr. R. Foye, Mail Stop 188A
1	Dr. R. J. Hayduk, Mail Stop 243
	National Aeronautics and Space Administration, Lewis Research Center, 21000 Brookpark Road, Cleveland, Ohio 44135
1	ATTN: Mr. S. S. Manson
1	Dr. J. E. Srawley, Mail Stop 105-1
1	Mr. W. F. Brown, Jr.
1	Mr. R. F. Lark, Mail Stop 49-3
1	Mr. M. H. Hirschberg, Head, Fatigue Research Section, Mail Stop 49-1
	Lockheed-Georgia Company, 86 South Cobb Drive, Marietta, Georgia 30063
1	ATTN: Materials & Processes Eng. Dept. 71-11, Zone 54
	National Bureau of Standards, U. S. Department of Commerce, Washington, D.C. 20234
1	ATTN: Mr. J. A. Bennett
1	Mechanical Properties Data Center, Belfour Stulen Inc., 13917 W. Bay Shore Drive, Traverse City, Michigan 49684
1	Mr. W. F. Anderson, Atomics International, Canoga Park, California 91303

No. of Copies	To
1	Dr. Thomas P. Rich, Department of Mechanical Engineering Texas A&M University, College Station, Texas 79602
1	Dr. Robert S. Shane, Shane Associates, Inc., 7821 Carrleigh Parkway, Springfield, Virginia 22152
	Midwest Research Institute, 425 Coker Boulevard, Kansas City, Missouri 64110
1	ATTN: Mr. G. Gross
1	Dr. J. Charles Grosskreutz, Asst. Dir. for Research, Solar Energy Research Institute, 1536 Cole Boulevard, Golden, Colorado 80401
1	Mr. A. Hurlich, Convair Div., General Dynamics Corp., Mail Zone 630-01, P. O. Box 80847, San Diego, California 92138
	Southwest Research Institute, 8500 Culebra Road, San Antonio, Texas 78284
1	ATTN: Dr. P. Francis
1	Dr. W. Baker
1	Mr. J. G. Kaufman, Alcoa Research Laboratories, New Kensington, Pennsylvania 15068
1	Mr. P. N. Randall, TRW Systems Group - 0-1/2210, One Space Park, Redondo Beach, California 90278
1	Dr. E. A. Steigerwald, TRW Metals Division, P. O. Box 250, Minerva, Ohio 44657
1	Dr. George R. Irwin, Department of Mechanical Engineering, University of Maryland, College Park, Maryland 20742
1	Mr. W. A. Van der Sluys, Research Center, Babcock and Wilcox, Alliance, Ohio 44601
1	Mr. B. M. Wundt, 2346 Shirl Lane, Schenectady, New York 12309
	Battelle Columbus Laboratories, 505 King Avenue, Columbus, Ohio 43201
1	ATTN: Mr. J. Campbell
1	Dr. G. T. Hahn
1	R. G. Hoagland, Metal Science Group
1	Dr. E. Rybicki
	General Electric Company, Schenectady, New York 12010
1	ATTN: Mr. A. J. Brothers, Materials & Processes Laboratory
	General Electric Company, Schenectady, New York 12309
1	ATTN: Mr. S. Yukawa, Metallurgy Unit
1	Mr. E. E. Zwicky, Jr.
	General Electric Company, Knolls Atomic Power Laboratory, P. O. Box 1072, Schenectady, New York 12301
1	ATTN: Mr. F. J. Mehringer
1	Dr. L. F. Coffin, Room 1C41-K1, Corp. R&D, General Electric Company, P. O. Box 8, Schenectady, New York 12301
	United States Steel Corporation, Monroeville, Pennsylvania 15146
1	ATTN: Mr. S. R. Novak
1	Dr. A. K. Shoemaker, Research Laboratory, Mail Stop 78
	Westinghouse Electric Company, Bettis Atomic Power Laboratory, P. O. Box 109, West Mifflin, Pennsylvania 15122
1	ATTN: Mr. M. L. Parrish
	Westinghouse Research and Development Center, 1310 Beulah Road, Pittsburgh, Pennsylvania 15235
1	ATTN: Mr. E. T. Wessel
1	Mr. M. J. Manjoine
1	Dr. Alan S. Tetelman, Failure Analysis Associates, Suite 4, 11777 Mississippi Ave., Los Angeles, California 90025
	Brown University, Providence, Rhode Island 02912
1	ATTN: Prof. J. R. Rice
1	Prof. W. N. Findley, Division of Engineering, Box D
1	Prof. P. C. Paris
	Carnegie-Mellon University, Department of Mechanical Engineering, Schenley Park, Pittsburgh, Pennsylvania 15213
1	ATTN: Dr. J. L. Swedlow
1	Prof. J. D. Lubahn, Colorado School of Mines, Golden, Colorado 80401
1	Prof. J. Dvorak, Civil Engineering Department, Duke University, Durham, North Carolina 27706
	George Washington University, School of Engineering and Applied Sciences, Washington, D.C. 20052
1	ATTN: Dr. H. Liebowitz
	Lehigh University, Bethlehem, Pennsylvania 18015
1	ATTN: Prof. G. C. Sih
1	Prof. R. Roberts
1	Prof. R. P. Wei
1	Prof. F. Erodgan
	Terra Tek, University Research Park, 420 Wakara Way, Salt Lake City, Utah 84108
1	ATTN: Dr. A. Jones
	P. R. Mallory Company, Inc., 3029 East Washington Street, Indianapolis, Indiana 46206
1	ATTN: Technical Library
1	Librarian, Material Sciences Corporation, Blue Bell Office Campus, Merion Towle House, Blue Bell, Pennsylvania 19422
	Massachusetts Institute of Technology, Cambridge, Massachusetts 02139
1	ATTN: Prof. B. L. Averbach, Materials Center, 13-5082
1	Prof. F. A. McClintock, Room 1-304
1	Prof. R. M. Pelloux
1	Prof. T. H. H. Pian, Department of Aeronautics and Astronautics
1	Prof. A. S. Argon, Room 1-306
1	Prof. J. N. Rossettos, Department of Mechanical Engineering, Northeastern University, Boston Massachusetts 02115
1	Prof. R. Greif, Department of Mechanical Engineering, Tufts University, Medford, Massachusetts 02155
1	Dr. D. E. Johnson, AVCO Systems Division, Wilmington, Massachusetts 01887
	University of Delaware, Department of Aerospace and Mechanical Engineering, Newark, Delaware 19711
1	ATTN: Prof. B. Pipes
1	Prof. J. Vinson
	Syracuse University, Department of Chemical Engineering and Metallurgy, 409 Link Hall, Syracuse, New York 13210
1	ATTN: Mr. H. W. Liu
1	Dr. V. Weiss, Metallurgical Research Labs., Bldg. D-6
1	Prof. E. R. Parker, Department of Materials Science and Engineering, University of California, Berkeley, California 94700
1	Prof. W. Goldsmith, Department of Mechanical Engineering, University of California, Berkeley, California 94720
	University of California, Los Alamos Scientific Laboratory, Los Alamos, New Mexico 87544
1	ATTN: Dr. R. Karpp
1	Prof. A. J. McEvily, Metallurgy Department U-136, University of Connecticut, Storrs, Connecticut 06268
1	Prof. D. Drucker, Dean of School of Engineering, University of Illinois, Champaign, Illinois 61820
	University of Illinois, Urbana, Illinois 61801
1	ATTN: Prof. H. T. Corten, Department of Theoretical and Applied Mechanics, 212 Talbot Laboratory
1	Prof. T. J. Dolan, Department of Theoretical and Applied Mechanics
1	Prof. J. Morrow, 321 Talbot Laboratory
1	Mr. G. M. Sinclair, Department of Theoretical and Applied Mechanics
1	Dr. T. Lardner, Department of Theoretical and Applied Mechanics
1	Prof. R. I. Stephens, Materials Engineering Division, University of Iowa, Iowa City, Iowa 52242
1	Prof. D. K. Felbeck, Department of Mechanical Engineering, University of Michigan, 2046 East Engineering, Ann Arbor, Michigan 48109
1	Dr. M. L. Williams, Dean of Engineering, 240 Benedum Hall, University of Pittsburgh, Pittsburgh, Pennsylvania 15260
1	Prof. A. Kobayashi, Department of Mechanical Engineering, FU-10, University of Washington, Seattle, Washington 98195
	State University of New York at Stony Brook, Stony Brook, New York 11790
1	ATTN: Prof. Fu-Pen Chiang, Department of Mechanics
	Denver Research Institute, 2390 South University Boulevard, Denver, Colorado 80210
1	ATTN: Dr. R. Recht
	Director, Army Materials and Mechanics Research Center, Watertown, Massachusetts 02172
2	ATTN: DRXMR-PL
1	DRXMR-WD
1	Author